EXPLORING STRUCTURAL REGULARITIES FOR ROBUST 3D RECONSTRUCTION OF URBAN SCENES

BY

ZIHAN ZHOU

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2013

Urbana, Illinois

Doctoral Committee:

Adjunct Associate Professor Yi Ma, Chair
Professor Thomas S. Huang
Professor Zhi-Pei Liang
Assistant Professor Derek W. Hoiem

Abstract

Driven by emergent needs in industrial applications such as film production, navigation and virtual reality, the problem of inferring 3D structures of urban scenes from 2D images has recently drawn a lot of interest in the computer vision community. Despite the extremely rich literature in multiple view geometry and structure from motion (SFM), reconstructing large-scale high-quality 3D urban models still remains a challenging problem.

A key feature of urban scene that differentiates it from other kinds of landscapes is the presence of strong structural regularities, such as planar surfaces, repetitive structures and all types of symmetries. While such regularities are largely ignored by existing SFM systems, in this thesis we demonstrate how they can be used to greatly facilitate 3D urban reconstruction as well as other related vision tasks.

In the first part of the thesis, we first look into the problem of structure and motion recovery directly from one or more large planes in the scene. We develop a new SFM method that generates high-quality reconstruction results in a short time, while avoiding several practical difficulties of conventional methods. Then, we show how the recovered planar structures can be seamlessly integrated into the current state-of-the-art video stabilization systems to obtain high-quality jitter-free videos in many challenging cases.

In the second part of the thesis, we focus on the structural regularities in visual data which give rise to a low-rank matrix structure, and develop a series of tools to recover them from images and videos. After reviewing the recent developments of convex optimization techniques for low-rank matrix recovery, we propose a novel 3D reconstruction approach based on a new class of global features called transform invariant low-rank textures (TILT).

We demonstrate the advantage of such global features over traditional local features in handling large-baseline images, occlusions, and repetitive patterns. In addition, we extend the tools from low-rank matrix recovery to harness the redundancy and temporal correlations among a large number of video frames, which leads to a novel method for generating clean textured models for street views.

For future work, my focus is on developing new methods for discovering complex structural regularities in urban scenes from large-scale visual data. I believe such methods would have a big impact in many modern industry applications in the near future.

To my wife

Zhenhui

for her Unbounded Love

And to my parents

Jin and Yan

for their Unconditional Support

Acknowledgments

Foremost, I would like to express my sincere gratitude to my advisor, Prof. Yi Ma, for his continuous support since the first day of my PhD study. His constant enthusiasm, immense knowledge and rigorous attitude towards computer vision research have deeply shaped my view of what scholarship, at its best, can be. Meanwhile, I am extremely thankful for his encouragement, patience and understanding throughout my journey to the PhD degree and beyond. Besides my advisor, I would also like to thank the rest of my thesis committee, Prof. Thomas S. Huang, Prof. Zhi-Pei Liang, and Prof. Derek W. Hoiem for their support, insightful comments and challenging questions.

Many thanks are due to Dr. Hailin Jin of Adobe Systems Inc., who offered me the summer internship opportunity (and much fun) in his group. Without him, much of this work would not have been possible.

I thank my fellow labmates and excellent collaborators, John Wright, Andrew Wagner, Shankar Rao, Arvind Ganesh, Yoav Sharon, Hossein Mobahi, Allen Yang, Kerui Min and Guangcan Liu for the inspiring discussions, for the sleepless nights we spent together before deadlines, and for all the fun we have had in the past five years. I would also like to thank my friends Jianchao Yang, Shen-Fu Tsai, Zhen Li, Xianbiao Shu, Liansheng Zhuang, Yigang Peng and Zhengdong Zhang, without whom my life at UIUC would be unthinkable.

Last but not the least, I would like to thank Zhenhui Li, who agreed to be my wife when I was in the most stressful stage of my PhD study. Her endless love has kept me going through every stage of the journey. Finally, I would like thank my parents, Jin Zhou and Yan Jia, for giving birth to me, and for years of unlimited support and understanding.

Table of Contents

Chapter 1 Introduction 1
1.1 Organization and Summary of Contributions 4

Chapter 2 Robust Plane-Based Structure From Motion 7
2.1 Introduction 7
2.1.1 Related Work 8
2.1.2 Contributions of this Work 9
2.2 Overview of the Method 11
2.3 Robust Plane Detection and Tracking 14
2.4 Plane-Based Self-Calibration 16
2.4.1 Self-Calibration with Constant Focal Length 16
2.4.2 Handling Varying Focal Length 18
2.5 Optimal Structure and Motion Recovery 19
2.6 Experiments 19
2.7 Conclusion 24

Chapter 3 Plane-Based Content-Preserving Warps for Video Stabilization 25
3.1 Introduction 25
3.1.1 Related Work 28
3.2 Overview of the Content-Preserving Warping Technique 29
3.3 Fast Piecewise Planar and Non-Planar Scene Segmentation for Videos 32
3.3.1 Multiple Plane Detection 32
3.3.2 A Markov Random Field Formulation for Video Segmentation 33
3.4 Plane-Based Stabilization 37
3.4.1 Quantitative Comparison of Two Warping Methods 38
3.5 Video Stabilization Results 40
3.6 Conclusion, Limitations, and Future Work 42

Chapter 4 Low-Rank Matrix Recovery via Convex Optimization 43
4.1 Introduction to Principal Component Pursuit 43
4.2 Stable Principal Component Pursuit 46
4.2.1 Assumption and Main Result 46
4.2.2 Relations to Existing Work 47
4.2.3 Notation and Outline of Analysis 48
4.2.4 Two Lemmas 50

4.2.5 Proof of Proposition 4 . . . 53
4.2.6 Simulations . . . 55
4.2.7 Discussion . . . 58

Chapter 5 Holistic 3D Reconstruction from Low-Rank Textures . . . 59
5.1 Introduction . . . 59
5.2 Geometry from One Facade of a Building . . . 61
5.3 Geometry from Intersecting Facades . . . 63
5.4 Segmenting Building Facades . . . 65
5.4.1 Compact Coding for Low-rank Textures . . . 67
5.4.2 Compression-Based Facade Segmentation . . . 70
5.5 Point-wise Matching of Building Facades . . . 71
5.6 Full 3D Reconstruction of Buildings . . . 73

Chapter 6 Low-rank Panoramas for Street View Videos . . . 77
6.1 Introduction . . . 77
6.1.1 Related Work . . . 81
6.2 Problem Formulation . . . 83
6.3 Robust Low-rank Panoramas via Convex Optimization . . . 86
6.3.1 Simulation on Synthetic Data . . . 87
6.3.2 Handling Non-uniform Sampling for Image Stitching . . . 90
6.4 Robust and Accurate Video Registration . . . 90
6.5 Experiments . . . 94

Chapter 7 Discussion and Conclusions . . . 97

References . . . 99

Chapter 1

Introduction

Reconstructing 3D structures of a scene from its 2D images has long been a central research topic in computer vision, with successful applications in many areas ranging from film production and virtual reality to navigation and robotics. Since this process typically involves estimating both 3D structure and camera motion at the same time, it is commonly known as *structure from motion* (SFM). More recently, largely driven by industrial applications such as Google Earth, Street View, and Microsoft's Bing Maps, there has been tremendous interest in building large-scale 3D models for *urban areas*. To meet the demands of such applications, significant progress about SFM techniques has been made in terms of the scalability and reliability [88, 98].

The conventional SFM approach to build a 3D model of a scene typically relies on detecting, matching, and triangulating a set of *feature points* (e.g., corners or SIFT features) across multiple camera views, which has been extensively studied in the past two to three decades. One great advantage of working with point features is that the system can be somewhat oblivious to the scene: the scene could be of any shape or texture as long as the structure is *general*, the motion is a single *rigid body* and texture is rich of *distinguishable feature points*.[1]

However, in practice, researchers have observed that urban scenes often exhibit strong structural regularities, such as planar surfaces, repetitive patterns, symmetries and self-symmetries. Intuitively, the presence of all types of regularities provides opportunities for

[1]There have been multiple parallel lines of work in studying 3D shape reconstruction for scenes that lack rich textures, using cues such as shape from shading and contours, etc.

(a) (b)

Figure 1.1: The presence of structural regularities poses significant challenges for conventional SFM systems. **(a)** Plane degeneracy. **(b)** Ambiguity in matching.

constraining and simplifying the reconstruction task. But rather surprisingly, such regularities actually pose significant challenges for conventional SFM systems:

- **Plane degeneracy.** The presence of a dominant plane is very common in man-made environments (Figure 1.1(a)). However, it violates the general structure assumption of traditional 3D reconstruction methods, and therefore often leads to ambiguous and even meaningless solutions.

- **Ambiguity in matching local features.** The fact that urban scenes are full of symmetry or self-symmetry, and repetitive patterns, makes the matching of local features across different views extremely difficult (Figure 1.1(b)). This problem could get even more drastic when the baseline between views is large, and/or the images are subject to occlusions and illumination changes.

Despite the above difficulties, it has been shown that structural regularities can greatly facilitate many 3D reconstruction and modeling tasks. If a method can take advantage of the presence of structural regularities in the scene, it is expected to achieve more efficient, robust, and accurate results. Consider the following two examples:

- **Handling textureless regions.** Existing methods for inferring 3D models of urban scenes typically require textured surfaces, and hence work poorly in textureless regions, such as the ground and building interiors. However, people have noticed that these

textureless regions often correspond to the large planar surfaces in the scene. Therefore, piecewise planar models have become popular in recent years for modeling indoor and outdoor urban scenes. For example, in the past two or three years, several piecewise planar stereo algorithms [97, 43, 44, 80] have been developed to produce the state-of-the-art mutli-view stereo results for urban scenes.

- **Robustness to perspective distortions, illumination changes and occlusions.** As a crucial part of the architectural scene reconstruction task, building textured geometric models for building facades from street view images and videos has drawn tremendous interest lately. However, this is not a trivial task at all, due to the presence of perspective distortions, illumination changes, as well as dynamic and unwanted foreground objects in the scene. To tackle these problems, traditional approaches often rely on local features and robust statistical techniques such as RANSAC, but only achieve limited success. Recently, it is demonstrated that holistic approaches [59, 64] which explore the scene regularities (e.g., repetitive patterns, symmetries) can obtain satisfactory results at a much higher level of variations in appearance and corruptions.

In view of the above challenges and opportunities, in this dissertation we develop a series of new tools for 3D reconstruction of urban scenes by exploiting various types of structural regularities in the scene including (1) piecewise planar structures, (2) symmetric or regular textures, and (3) linear correlations between images. We show that our methods not only avoid the aforementioned difficulties of conventional methods (e.g., plane degeneracy, ambiguity in matching local features), but also outperform existing methods in handling textureless regions, dynamic foregrounds, occlusions, reflections and illumination changes in the image sequences.

1.1 Organization and Summary of Contributions

This thesis can be divided into two parts. The first part (Chapter 2 - 3) is focused on the detection, reconstruction of **piecewise planar structures** in the scene, and their applications in 3D modeling. In this part, we still rely on feature tracks to detect planes in the scene. In the second part (Chapter 4 - 6), we show how the symmetric and regular patterns in a single image and the linear correlations between multiple images can be both captured by a **low-rank structure** model. By leveraging powerful high-dimensional convex optimization tools from compressive sensing of sparse signals and low-rank matrix recovery, we develop new holistic approaches for reconstructing geometric and textural models of urban scenes without using local features.

In Chapter 2, we introduce a new approach to structure and motion recovery directly from one or more large planes in the scene. When such a plane exists, we demonstrate how to automatically detect and track it robustly and consistently over a long video sequence, and how to efficiently self-calibrate the camera using the homographies induced by this plane. We build a complete structure from motion system which does not use any additional off-the-plane information about the scene, and show its advantage over conventional systems in handling two important issues in real world videos, namely, the plane degeneracy and the dynamic foreground problems.

In Chapter 3, we further demonstrate how the planes reconstructed via SFM can be used to substantially boost the performance of existing methods for other 3D modeling tasks, such as video stabilization. Particularly, we investigate a newly developed image deformation technique called content-preserving warping (CPW), which is shown to produce the state-of-the-art video stabilization results in many challenging cases. Since CPW solely relies on the tracked feature points to guide the warping, it works poorly in large textureless regions, such as ground and building interiors. To overcome this limitation, we present a hybrid approach for novel view synthesis, observing that the textureless regions often correspond to large

planar surfaces in the scene. Specifically, given a jittery video, we first segment each frame into piecewise planar regions as well as regions labeled as non-planar using Markov random fields. Then, a new warp is computed by estimating a single homography for regions belong to the same plane, while inheriting results from CPW in the non-planar regions. By seamlessly integrating the information about planar structures into the stabilization framework, our new method is able to generate high-quality jitter-free videos in a variety of practical scenarios.

In Chapter 4, we briefly review some of the recent developments in the field of low-rank matrix recovery. In particular, we describe the newly proposed Principal Component Pursuit (PCP) method [18], which utilizes a convex program to recover the low-rank matrix L_0 from corrupted observations $M = L_0 + S_0$, where S_0 is a sparse error matrix. It is shown in [18] that under quite broad conditions, the convex program exactly recovers L_0 even if a constant fraction of entries in M are grossly corrupted. We extend this result to the case where the observation matrix M is also subject to small entry-wise noise: $M = L_0 + S_0 + Z_0$, and prove that under the same conditions as PCP, a relaxed version of the convex program gives a stable estimate of L_0 and S_0. These results form the basis for the 3D reconstruction techniques we will introduce in the next two chapters.

In Chapter 5, we introduce a new approach to reconstructing accurate camera geometry and 3D models for urban structures in a holistic fashion, i.e., without relying on extraction or matching of traditional local features such as points and edges. Instead, we use a new class of image features called the transform invariant low-rank textures (TILT) [118], which extend PCP to handle transformations in the image domain. Modern high-dimensional optimization techniques enable us to accurately and robustly recover precise and consistent camera calibration and scene geometry from single or multiple images of the scene. We demonstrate how to construct 3D models of large-scale buildings from sequences of multiple large-baseline uncalibrated images that conventional SFM systems do not apply.

While Chapter 5 is focused on the low-rank structures within a single image, in Chapter 6 we explore the low-rank structures among multiple video frames to generate clean street

view panoramas from videos. We formulate the problem as one of robust recovery of a low-rank matrix from highly incomplete, corrupted, and deformed measurements (the video frames). In particular, we show how the proposed method can effectively remove severe occlusions or corruptions (caused by trees, cars, or reflections, etc.), automatically and robustly establish pixel-wise accurate registration among all the video frames, and finally obtain street panoramas that have very clean global appearance and very accurate global geometry.

Chapter 2

Robust Plane-Based Structure From Motion

One of the most prominent characteristic of man-made environments is the presence of one or more (relatively large) planes. Rather surprisingly, such structural regularity has been largely ignored by existing general-purpose SFM systems. In this chapter, we develop a reliable SFM system that can explicitly take advantage of the planar structures in the scene, and demonstrate its effectiveness in handling various challenging cases in real 3D reconstruction applications.

2.1 Introduction

It is well known that planar structures encode very important geometric information about the scene, which can be used to greatly facilitate the 3D reconstruction task [7, 83]. Besides their benefits for reconstruction, the recovered planes are also of great interest to people working on other 3D modeling applications, as it provides a compact, abstract representation of the architectural objects.

In order to leverage information about planar structures for our task, the first task obviously is to automatically detect such a plane in the scene from a given image sequence. This turns out to be not so trivial at all. For instance, one may attempt to detect planes between adjacent image pairs and combine the detection result across multiple pairs. However, since the camera motion between two adjacent frames is usually small, the detection result is very sensitive to noise. Further, the planes detected in different pairs of images may not be consistent with each other. Another practical difficulty is the presence of dynamic

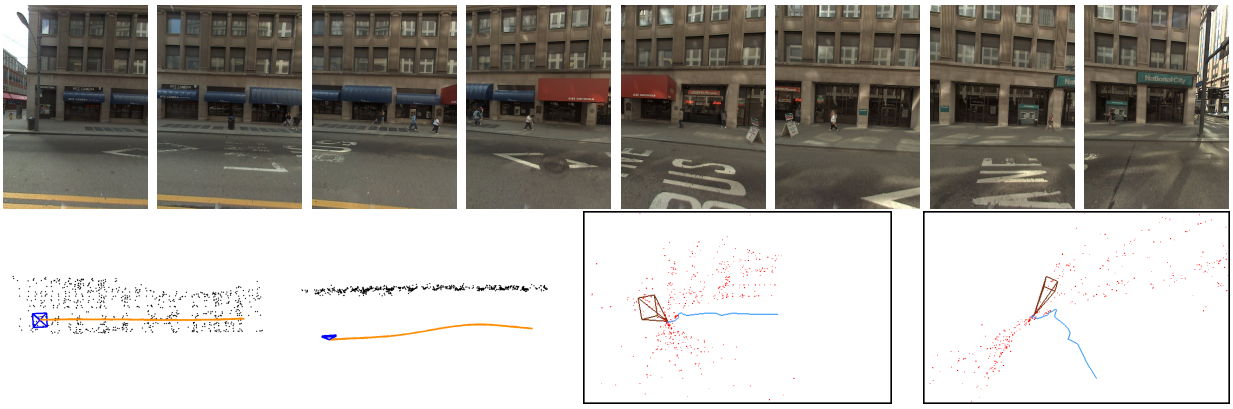
foreground in the scene. In fact, large majority of commercial or consumer videos consist of one or more moving objects, violating the single rigid body requirement, see Figures 2.2 and 2.3 for examples. Those objects, if not properly handled, could lead to huge detection and reconstruction errors in the final results. In this chapter, we assume that for most cases of interests, a relatively dominant plane, if existing in the scene, belongs to a static background, and the foreground consists of out-of-plane structures and possibly other independently moving objects. Our goal is hence to robustly detect the plane and accurately recover the static part of the scene, despite severe corruption by dynamic outliers.

2.1.1 Related Work

Prior knowledge about the scene planes has been explored before for SFM problem. For instance, [7] uses user-provided geometry about a piecewise planar scene to constrain the estimation of structure and motion parameters. [83] shows that the relationship between uncalibrated cameras and 3D scene points is linear with a known reference plane, and can be solved simultaneously via a linear algorithm. However, these methods require users to provide necessary information about the planes.

The problem of plane degeneracy in multi-view structure from motion has also been previously addressed. Several papers have tried to detect the degenerated frames and exclude them from the initial projective reconstruction by either fitting an average planar homography [87] between two frames or using some other statistic measures [107, 89]. Alternatively, [29] proposes a RANSAC-based algorithm for robust estimation of the epipolar geometry. These methods often substantially complicate the SFM system, and are not always reliable in practice, as noticed by [117]. Also, these methods assume the existence of enough out-of-plane structure, at least in certain part of the video, which may be unrealistic for many practical scenarios.

A popular method for detecting planar structure between two frames is to use RANSAC [39]. In this chapter, we show how to extend this method to produce consistent plane models



(a) (b) (c) (d)

Figure 2.1: "Google Street View" example. **Top row:** Eight snapshots of the input video from Google Street View taken by a smoothly moving camera mounted on the car. **(a) and (b)** Frontal and top view of the reconstruction results of our plane-based SFM algorithm. **(c) and (d)** Incorrect reconstruction result from one of the state-of-the-art systems [117].

over long video sequences. Recently, [90] proposes a model selection method for multiple-frame plane detection using the Minimal Description Length (MDL) principle. While it focuses on discovering multiple plane models simultaneously, its robustness to gross outliers is unknown.

Finally, with the seminal work by Triggs [108], various approaches for camera self-calibration from a planar scene have been developed over the past decade. However, many of these methods require additional assumptions on the data (e.g., fronto-parallelism of the key image) or user input to initialize the local optimization algorithm [74, 53, 77]. Assuming that only the constant focal length is unknown, a global solution is derived in [13]. But this method does not scale beyond a small number of views, and hence is not suitable for our purpose.

2.1.2 Contributions of this Work

In this chapter, we propose a novel and complete automatic SFM system specifically designed to exploit the useful properties of scene planes, meanwhile avoiding the aforementioned difficulties of conventional methods. We show how to automatically and robustly detect a scene

Figure 2.2: Three consecutive frames of the "Google Street View" sequence with the detected plane (building facade) using our method. Green dots correspond to the inlying points on the plane, red dots correspond to outliers. Note the outliers on the window glasses due to reflection.

plane (if present) and obtain accurate information about the cameras and structures directly from the plane, *without using any additional off-the-plane information about the scene*. Our method can handle multiple planes in the scene in a unified manner, and there is no need for images in the sequence to share a common plane (see the "Wall" and "Office Desk" sequences in Figure 2.5). As a result, our system produces clean, simple and visually plausible models for various challenging commercial or consumer videos on which conventional SFM systems often fail.

Figure 2.1 shows an example of successful reconstruction of a challenging sequence captured by Google Street View[1] using our method. Such sequences are of great importance to the computer vision community nowadays due to the increasing interest in building large-scale 3D models for urban area from the industry. However, conventional SFM systems often perform very poorly on them because (1) most of the tracked point trajectories lie on a plane (i.e., the building facade) in the scene and (2) there exists a significant amount of outliers due to the reflection of window glass, moving objects, etc. As one can see in Figure 2.1, the reconstruction result by one of the state-of-the-art SFM systems [117] is obviously wrong.

The success of our system relies on several technical improvements over existing methods and systems, with the following notable advantages:

- We develop a novel method called TRASAC (TRAjectory SAmpling Consensus) for

[1] www.google.com/streetview

robust plane detection and tracking from video sequences. This method generalizes the classical two-frame RANSAC to estimate consistent plane models across multiple views, and has a very high breakdown point to gross outliers. This ensures that our method is much more robust than conventional SFM methods which utilize epipolar geometry for outlier rejection or two-frame RANSAC.

- We propose a fully automatic plane-based self-calibration approach, which is fast, easy to implement and yet able to reliably handle practical sequences that have significant varying focal lengths. This makes our system very robust to initialization of the camera calibration and significantly enhances its applicability to commercial or consumer videos.

- Another advantage of our method is that the motion parameters for all the cameras are initialized globally. In contrast, most traditional SFM methods such as [41, 88, 117] employ an incremental method, i.e., they solve for progressively larger sets of images. Incremental methods are known to be sensitive to the initialization and amenable to local minima. Furthermore, our method is significantly more efficient than existing work for obtaining global initialization using a hybrid discrete-continuous optimization method [30].

2.2 Overview of the Method

Before introducing our method, we review some notations and backgrounds of the multi-view geometry [54, 72]. Supposing a rigid scene is viewed by N cameras, we use $K_i \in \mathbb{R}^{3\times3}$ to denote the intrinsic matrix of the i-th camera. Without loss of generality, we choose the world coordinate frame to be the camera frame of the first camera, and use $R_i \in SO(3)$ and $\boldsymbol{t}_i \in \mathbb{R}^3$ to denote the Euclidean transformation from the world coordinate frame to the i-th camera frame.

For a piecewise planar scene with P planes, we assume that a 3D plane π_k $(1 \leq k \leq P)$ has coordinates $\pi_k = (\boldsymbol{n}_k, d_k)^T$ with respect to the world coordinate frame, where $\boldsymbol{n}_k$ is the unit normal vector and $d_k > 0$ denotes the distance from the plane to the world origin. Therefore, for any point $X \in \mathbb{R}^3$ on it we have $\boldsymbol{n}_k^T X = d_k$.

Consider the situation in which we observe a set of trajectories $\mathcal{T} = \{T_j\}_{j=1}^M$ of M feature points. For each T_j, let p_j and q_j $(1 \leq p_j < q_j \leq N)$ denote its starting and ending frames, respectively. We can therefore write $T_j = \{\boldsymbol{x}_j^i\}_{i=p_j}^{q_j}$, where $\boldsymbol{x}_j^i \in \mathbb{P}^2$ is the homogeneous coordinates of the j-th point as seen by the i-th camera. We also use $\mathcal{T}^{ab} = \{T_j \in \mathcal{T} : p_j \leq a, q_j \geq b\}$ to represent the set of trajectories which span the a-th and b-th frames.

Finally, if a tracked point lies on π_k, the coordinates of the first frame and the i-th frame are related by a planar homography $\boldsymbol{x}_j^i = H_i \boldsymbol{x}_j^1$ where H_i can be written as:

$$H_i \simeq K_i (R_i + \boldsymbol{t}_i \boldsymbol{n}_k^T / d_k) K_1^{-1}, \tag{2.1}$$

with the symbol $\simeq$ meaning "equality up to a scale."

Our approach takes the feature point trajectories obtained by any standard tracking algorithm as input. To measure the fitness of a plane model to a trajectory T_j, we use the sum of the squares of the standard Euclidian image distance in the i-th image, $\|\boldsymbol{x}_j^i - H_i \boldsymbol{x}_j\|^2$, for all i's between p_j and q_j. Note that here we use $\boldsymbol{x}_j$ as the (to be estimated) true feature point location in the first frame. This is different from $\boldsymbol{x}_j^1$, the (possibly noisy) 2D measurement of the same quantity.

Our goal is then to partition all the trajectories into groups, each corresponding to a plane in the scene, plus a set of trajectories which are labeled as *outliers*. We emphasize that an outlier may either come from non-planar structures of a static scene (e.g., trees), or dynamic foreground objects (e.g., moving cars). Defining S_k as the set of indices of the trajectories which belong to the k-th plane, and S_0 as the set of outlying trajectories, we can now formulate our structure and motion recovery problem as minimizing the following

Figure 2.3: Selected frames of the "Beach" sequence with classified trajectories using TRASAC. Green: inliers. Red: outliers.

geometric error function:

$$\sum_{k=1}^{P}\sum_{j\in S_k}\sum_{i=p_j}^{q_j}\|\boldsymbol{x}_j^i - K_i(R_i + \frac{\boldsymbol{t}_i}{d_k}\boldsymbol{n}_k^T)K_1^{-1}\boldsymbol{x}_j\|^2 + \sum_{j\in S_0}\sum_{i=p_j}^{q_j}\eta^2, \tag{2.2}$$

where η is the penalty for labeling a trajectory as an outlier.

In order to minimize this nonlinear function, we use an alternating method, which iterates between updating the plane models and assigning each trajectory to current plane candidates. Like other local methods, a set of good initial values of the unknowns are crucial for the algorithm to converge to the desired solution. In this chapter, we propose to find such a good initialization using a two-stage approach. First, we detect and track each plane using a robust algorithm, yielding a set of inter-image homographies induced by the planes (Section 2.3). Second, we develop a plane-based self-calibration method which takes the homography matrices as the input and outputs the structure and motion parameters (Section 2.4). This is followed by the aforementioned alternating scheme which refines all the parameters (Section 2.5). We illustrate the performance of our method in Section 2.6 and conclude our discussion in Section 2.7.

2.3 Robust Plane Detection and Tracking

In this section, we describe a novel method called TRASAC, which is a generalization of the RANSAC estimator, for detecting and tracking *one* plane in the video sequence. To obtain all the planes one can simply apply this method sequentially by removing the inliers of the current plane after each iteration.

The novelty of our method is that instead of independently sampling point correspondences between every two frames, it directly samples the feature point trajectories. By doing so, we assume that if a trajectory is classified as an inlier within any pair of frames, it remains as an inlier to the same plane for all the other frames it spans. Compared to the two-frame RANSAC, the advantage of our method is two-fold: First, it directly generates a consistent plane model over the entire sequence – no linking is needed as a post-processing step. Second, it enables us to use only trajectories with *known* membership to estimate the homographies induced by the same plane in the rest of the frames. In this way, we derive an efficient algorithm with very high tolerance to (possibly dominant) outliers in the scene.

We now discuss our method in full detail. Since our method is based on sampling consensus, it consists of multiple trials of the same procedure followed by a selection of the best result from these trials. We first describe the procedure of one trial, which contains two steps (Step 1 and 2 below). Note that given an input sequence, our method operates in an incremental manner, processing two adjacent frames at a time. Therefore, for each trial, we maintain the sets of trajectories which are classified as inliers and outliers, $\mathcal{T}_{in}$ and $\mathcal{T}_{out}$, respectively. They are both empty at the beginning, and expanded accordingly after processing each image pair.

Step 1: Random sampling. Given an input sequence of N frames $\{F_i\}_{i=1}^{N}$, we form $N-1$ pairs of adjacent frames $\mathcal{C} = \{(F_1, F_2), (F_2, F_3), \ldots, (F_{N-1}, F_N)\}$. Our algorithm starts with a randomly chosen pair in $\mathcal{C}$, say (F_{i-1}, F_i). Then, a putative plane model between these two frames is generated using a random minimum subset of samples. More precisely, 4 randomly

Algorithm 1 (TRASAC)

Input: A set of M trajectories $\mathcal{T}$ over N frames. A distance threshold ϵ.
repeat for n **trials:**
  Select a random pair of frames (F_{i-1}, F_i) from $\mathcal{C}$.
  Select a random sample of four trajectories from $\mathcal{T}^{(i-1)i}$ and compute the homography $H_{(i-1)i}$.
  Classify each $T_j \in \mathcal{T}^{(i-1)i}$ into $\mathcal{T}_{in}$ or $\mathcal{T}_{out}$ according to $H_{(i-1)i}$.
  while not all pairs in $\mathcal{C}$ are processed
    Select a new pair of frames (F_{k-1}, F_k).
    if $|\mathcal{T}_{in} \bigcap \mathcal{T}^{(k-1)k}| \leq 4$; **break**; **end if**
    Compute $H_{(k-1)k}$ using two-frame RANSAC estimation from trajectories in $\mathcal{T}_{in} \bigcap \mathcal{T}^{(k-1)k}$.
    Classify all the unclassified trajectories in $\mathcal{T}^{(k-1)k}$ into $\mathcal{T}_{in}$ or $\mathcal{T}_{out}$ according to $H_{(k-1)k}$.
  end while
end repeat
Choose the set of homographies $\{H_{(i-1)i}\}_{i=2}^{N}$ from the trial with the largest number of inliers $|\mathcal{T}_{in}|$.
Compute the homography between the first and the i-th frame recursively using $\{H_{(i-1)i}\}_{i=2}^{N}$: $H_1 = I_{3\times 3}, H_i = H_{(i-1)i}H_{i-1}, i = 2, \ldots, N$.
Output: A set of inter-image homographies $\{H_i\}_{i=1}^{N}$.

chosen trajectories in $\mathcal{T}^{(i-1)i}$ are used to estimate a homography matrix $H_{(i-1)i}$. Then, given a fixed threshold ϵ, we classify each trajectory $T_j \in \mathcal{T}^{(i-1)i}$ into $\mathcal{T}_{in}$ or $\mathcal{T}_{out}$ by comparing the projection error $\|\boldsymbol{x}_j^i - H_{(i-1)i}\boldsymbol{x}_j^{i-1}\|$ with ϵ.

Step 2: Computation of the consensus. Next, we choose a new pair of frames which is adjacent to the previous pair, say (F_i, F_{i+1}),[2] and compute the set of trajectories in $\mathcal{T}^{i(i+1)}$ which have already been labeled as inliers, i.e., $\mathcal{T}_{in} \bigcap \mathcal{T}^{i(i+1)}$. These trajectories are then used as candidates to estimate the homography $H_{i(i+1)}$. In the ideal case, any 4 or more samples from this set should do the job equally well because they are all inliers. However, to ensure the estimation quality in the presence of image noise, we generate a small number of model hypotheses and select the one with the largest number of inliers. We repeat this step for each pair of adjacent frames, until all the frames are processed or there are not enough inliers to proceed.

Selection of the best model. After repeating Steps 1 and 2 for enough times, the plane

[2]The other adjacent pair is (F_{i-2}, F_{i-1}). We do not make any preference among these two choices.

model (i.e., a set of homographies) estimated from the trial with the largest total number of inliers across the entire sequence is kept as the output.

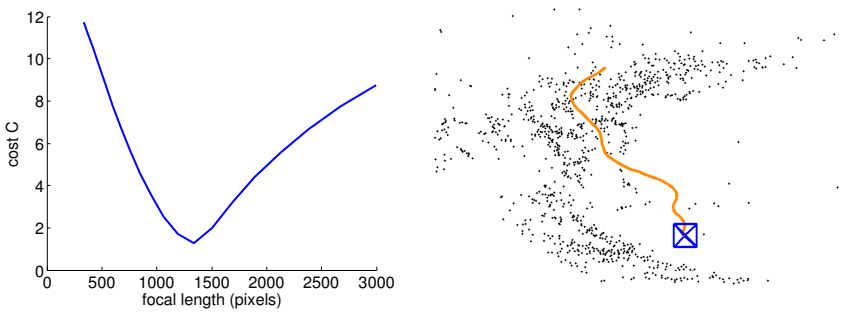
We summarize the complete procedure as Algorithm 1. The only parameter for our method is the distance threshold ϵ. Since our goal is to detect those large scene planes, we find that a fixed value $\epsilon = 4$ (pixels) works well enough in practice. Figure 2.3 shows an example of the detected plane (the ground) in the "Beach" sequence by TRASAC. As one can see, the plane detected by our method is consistent despite large number of outliers in certain frames.

2.4 Plane-Based Self-Calibration

In this section, we discuss how to self-calibrate a camera using only the set of homographies $\mathcal{H} = \{H_i\}_{i=1}^N$ induced by a scene plane. We assume the camera to have a zero pixel skew and known aspect ratio, which is true for most modern digital cameras. We also assume that the principal point coincides with the image center, as the error introduced by this approximation is normally well within the region of convergence of the subsequent nonlinear optimization. As a result, the self-calibration problem is reduced to finding the focal length for each frame. Inspired by the work of [46], we propose to enumerate the inherently bounded space of focal lengths and examine the tentative metric reconstruction produced by each sample. In the rest of this section, we first describe our method for the constant focal length case in detail. Then we will show how to generalize this method to handle varying focal length.

2.4.1 Self-Calibration with Constant Focal Length

Our self-calibration method is based on two important observations. First, if the focal length f (or equivalently the matrix K) is given, then there are at most two physically possible solutions for a decomposition of any H into parameters $\{R, \tilde{\boldsymbol{t}}, \boldsymbol{n}\}$ where $\tilde{\boldsymbol{t}} = \boldsymbol{t}/d$ (see e.g. [72]). Second, the space of possible values of f is inherently bounded by the finiteness of the



(a) (b)

Figure 2.4: (a) Score C as a function of f for the "Beach" sequence. It is minimized at the true focal length. (b) Reconstruction result.

acquisition devices. We assume $f \in [0.3f_0, 3f_0]$ where f_0 is defined as the sum of half width and half height of the image and propose the following two-stage method:

1. Given a guess on f, compute the plane normal $\boldsymbol{n}$ from the homography induced by any two frames.[3] This yields at most two physically possible normals. For each $\boldsymbol{n}$, estimate $\{R_i, \tilde{\boldsymbol{t}}_i\}_{i=2}^N$ for all cameras.

2. Enumerate the space of focal length (a subset of $\mathbb{R}$) and score each focal length f based on how well the recovered structure and motion parameters fit the homographies.

The best solution is then obtained according to the scores. We now elaborate each step in detail.

Planar homography decomposition. Given an estimate for the focal length, we can compute the Euclidean homography matrix as: $\hat{H}_i = K^{-1}H_iK$. $\hat{H}_i$ is related to $\{R_i, \tilde{\boldsymbol{t}}_i, \boldsymbol{n}\}$ as follows:

$$\hat{H}_i = \lambda_i(R_i + \tilde{\boldsymbol{t}}_i\boldsymbol{n}^T). \tag{2.3}$$

It turns out that there are only four solutions for decomposing $\hat{H}_i$ to $\{R_i, \tilde{\boldsymbol{t}}_i, \boldsymbol{n}\}$. The positive depth constraint can be imposed to reduce the number of physically possible solutions to two. We refer the reader to [72] for more details.

[3] In this work, we always choose the homography H_N between the first frame and the last frame for computing $\boldsymbol{n}$.

Estimation of the focal length. As mentioned before, our self-calibration algorithm determines the focal length f by enumerating all of its possible values and checking how well the resulting camera parameters $\{R_i, \tilde{\boldsymbol{t}}_i\}_{i=2}^N$ and plane normal $\boldsymbol{n}$ fit the homographies $\{\hat{H}_i\}_{i=2}^N$ where $\hat{H}_i = K^{-1}H_iK$. Once a set of parameters are obtained for a given f, there are several ways to score them. In this chapter, we adopt the cost function used in [74], which compares the normalized difference of the two non-zero singular values σ_i^1 and σ_i^2 ($\sigma_i^1 \geq \sigma_i^2$) of the matrix $\hat{H}_i\widehat{\boldsymbol{n}}$:

$$C = \sum_{i=2}^{N} \frac{\sigma_i^1 - \sigma_i^2}{\sigma_i^1}. \tag{2.4}$$

The computational complexity of our self-calibration algorithm is linear in the number of samples of f. Figure 2.4(a) shows a plot of the score as a function of focal length for the "Beach" sequence. As one can see, the correct focal length can be easily determined as the minimizing point on the curve. Once the camera is calibrated, the camera motion and scene points can be recovered as shown in Figure 2.4(b).

2.4.2 Handling Varying Focal Length

Our method can be easily generalized to handle the varying focal length case. Instead of sampling $f \in \mathbb{R}$, we sample all possible values of $(f_1, f_N) \in \mathbb{R}^2$ (the first and last cameras are chosen for convenience) and compute the plane normal $\boldsymbol{n}$ as described before. Compared to the constant focal length case, the extra work required is to compute the focal length for other images $f_2, \ldots, f_{N-1}$. We note that $K_i^{-1}H_iK_1$ has to preserve the length of any vectors inside the subspace perpendicular to $\boldsymbol{n}$ (see details in [72]). Leting $\boldsymbol{u}, \boldsymbol{v}$ be two unit vectors in that subspace, the length constraint dictates

$$\|K_i^{-1}H_iK_1\boldsymbol{v}\|^2 = \|K_i^{-1}H_iK_1\boldsymbol{u}\|^2. \tag{2.5}$$

Equation (2.5) is a linear equation in f_i^2 which can be easily solved to obtain f_i.

2.5 Optimal Structure and Motion Recovery

With a good initialization of all parameters, we solve the global optimization problem (2.2) using an alternating algorithm. On one hand, given the labeling $\{S_k\}_{k=0}^P$, (2.2) becomes:

$$\begin{aligned}&\min f(\boldsymbol{x}_j, K_i, R_i, \boldsymbol{t}_i, \boldsymbol{n}_k, d_k)\\ &= \sum_{k=1}^{P}\sum_{j\in S_k}\sum_{i=p_j}^{q_j} \|\boldsymbol{x}_j^i - K_i(R_i + \boldsymbol{t}_i\boldsymbol{n}_k^T/d_k)K_1^{-1}\boldsymbol{x}_j\|^2,\end{aligned}$$

which can be solved via the Levenberg-Marquardt (LM) method. On the other hand, given the structure and motion parameters, we can update the index sets $\{S_k\}_{k=0}^P$. For the trajectory T_j, let

$$f_j(k) = \sum_{i=p_j}^{q_j} \|\boldsymbol{x}_j^i - K_i(R_i + \boldsymbol{t}_i\boldsymbol{n}_k^T/d_k)K_1^{-1}\boldsymbol{x}_j\|^2.$$

We assign T_j to class k^* using the following rule:

$$k^* = \begin{cases} 0 & \text{if } \min_k f_j(k) > (q_j - p_j + 1)\eta^2 \\ \arg\min_k f_j(k) & \text{otherwise} \end{cases}$$

Full 3D reconstruction. To recover the full 3D structure, we back-project all the points on the plane to obtain their 3D positions. In addition, we can triangulate the positions of the off-the-plane points. We employ the standard 3D bundle adjustment to get the optimal estimates of all structure and motion parameters.

2.6 Experiments

We have tested our algorithm on more than 50 video sequences captured by a variety of cameras. These sequences cover a wide range of scenes with one or more large planes, from both natural and indoor/outdoor man-made environments. In the section, we report the reconstruction results of our method on several representative examples, which are shown in

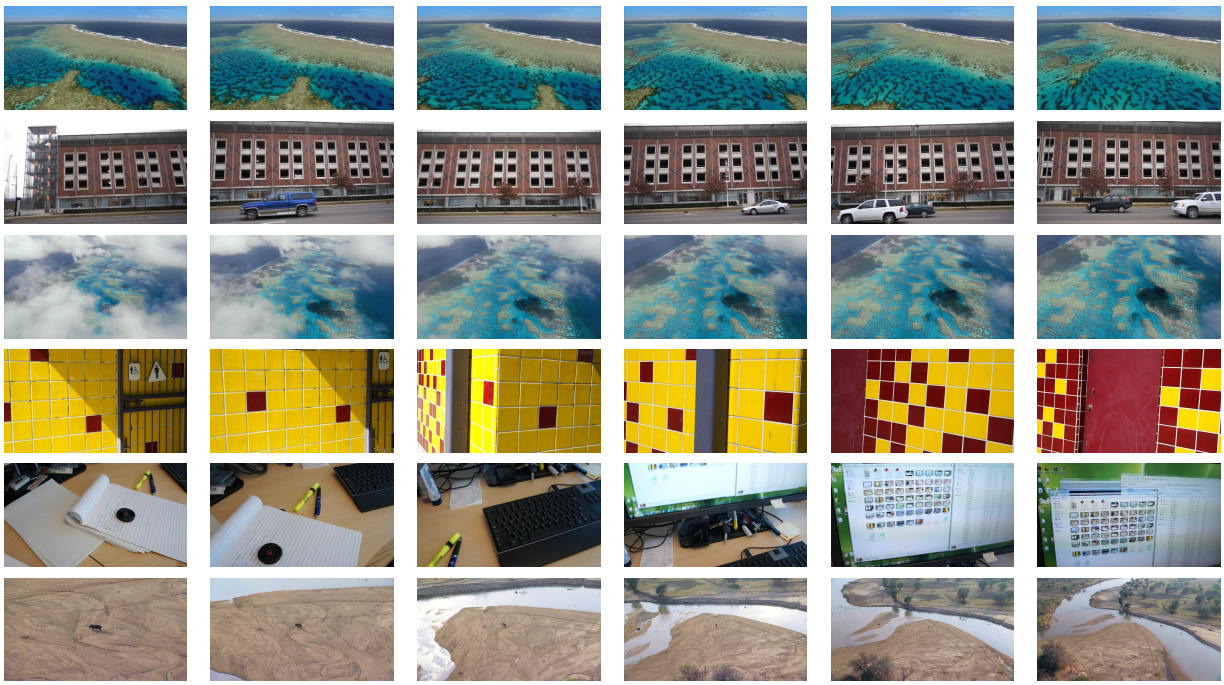


Figure 2.5: Snapshots of several testing sequences. **From top to bottom:** "Seashore", "Street", "Shallow Sea", "Wall", "Office Desk" and "Lonely Hippo".

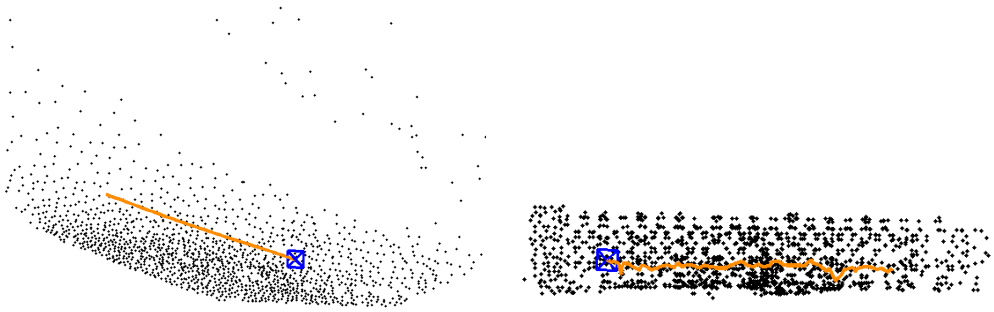
Figure 2.5. In terms of speed, for a typical sequence such as "Beach" with 660 frames, our system chooses 44 keyframes and reconstructs 1479 3D points, which takes about 50 seconds on a desktop PC with Intel Xeon 2.67GHz CPU and 24GB memory.

To better understand the reconstruction quality and the advantage of our method, we further compare our method against one of the state-of-the-art general-purpose SFM system, ACTS [117]. We have also tested Bundler [98] and Voodoo Camera Tracker[4] on these sequences. However, Bundler is designed for unordered large-baseline images and computes point correspondences between each pair of images, and hence is very inefficient for our purpose. Also, it assumes known camera intrinsic parameters. For Voodoo, we found that its performance is generally worse than ACTS; hence we omit its results here.

According to the performance of ACTS, we roughly partition the test sequences into two categories. The first category consists of planar scenes with no or little 3D structure throughout the entire sequence, whereas the second category contains videos with certain

[4]www.digilab.uni-hannover.de/docs/manual.html

Figure 2.6: Some augmented images of the "Beach" sequence using the reconstruction result obtained by our method.

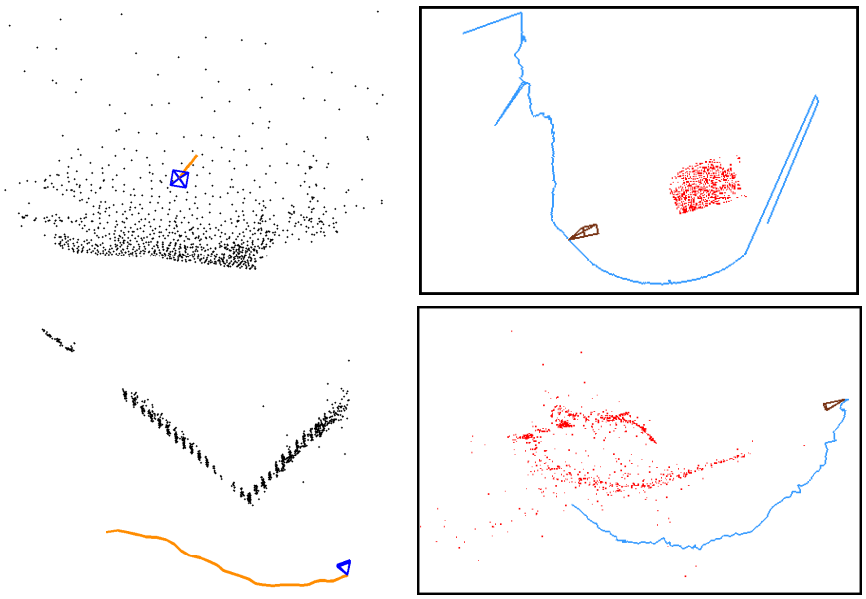


(a) Seashore (b) Street

Figure 2.7: Some reconstruction results of our method.

3D structure in at least a fraction of the frames (e.g., the "Beach" and "Office Desk" sequences). As expected, while sequences in the first category are considered easy to our method, ACTS performs poorly on them, generating incomplete or obviously wrong results. For the second category, ACTS is able to obtain reasonable solutions, thanks to its ability to detect key frames with enough 3D structures for initialization. For these sequences, we further demonstrate the reconstruction quality of our method by inserting virtual objects to the videos.

The "Beach" sequence. This is a representative example with both large dynamic foreground (sea waves, running people) and planar scene structure (Figure 2.3). We have already seen the reconstruction result of our method in Figure 2.4(b). Here, we further examine the



(a) our method (b) ACTS

Figure 2.8: Comparison of reconstruction results. **Top row:** The "Shallow Sea" sequence. **Bottom row:** The "Wall" sequence.

reconstruction result of our method by augmenting the video with a synthetic object. As one can see in Figure 2.6, the castle in our result remains firmly registered to the scene, implying the reconstruction by our method is very accurate.

The "Seashore" sequence. This sequence is taken by an aerial camera moving forward along the seashore. Because the scene is completely flat, ACTS crashes on this example. The reconstruction result of our method is shown in Figure 2.7(a).

The "Street" sequence. This is an example of planar scene in man-made environments with dynamic foregrounds (i.e., cars). The planar structure and camera motion are easily obtained by our method, as shown in Figure 2.7(b), while ACTS generates completely wrong result.

The "Shallow Sea" sequence. This is another example of a planar scene with large dynamic foreground (i.e., the clouds). In this sequence the camera is smoothly moving forward, which is correctly recovered by our method, as shown in Figure 2.8. However, ACTS fails in this case possibly due to lack of static 3D structure in the scene.

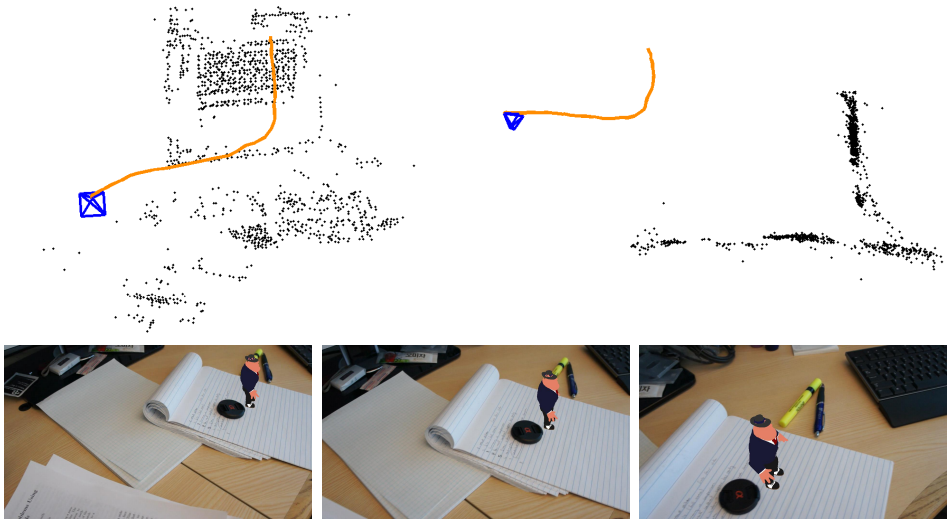


Figure 2.9: Reconstruction results of the "Office Desk" sequence. **Top row:** Two views of the result of our method. **Bottom row:** Augmented frames by our method.

The "Wall" sequence. We use this somewhat extreme example to test the ability of both systems in handling multiple planes. As one can see in Figure 2.8, the structure recovered by our system is very accurate, with a clean right angle between the two walls. In contrast, ACTS generates incorrect structure in this case.

The "Office Desk" sequence. This scene also contains two large planes, the desk and the computer monitor. In addition, as one can see in Figure 2.9, the synthetic object in our method's augmented video remains very steady throughout the sequence. This further evidences the advantage of using information encoded by scene planes for accurate reconstruction.

The "Lonely Hippo" sequence. Lastly, we test our method on a sequence with a smoothly zooming-out camera. It is very challenging in that the focal length changes by a factor of 8 between the first frame and the last frame. Figure 2.10 shows the estimated focal lengths as well as the reconstruction result by our method, verifying its effectiveness in handling varying focal length.

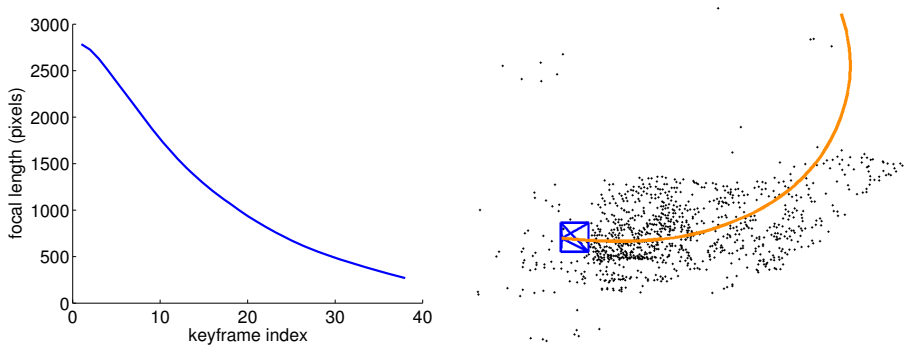


Figure 2.10: The "Lonely Hippo" sequence with varying focal length. **Left:** Estimated focal lengths. **Right:** Reconstruction result.

2.7 Conclusion

In this chapter, we have proposed a novel and complete SFM system which produces very accurate reconstruction result by directly analyzing the geometry information encoded by large scene planes. The system consists of two main components, namely, a new method to detect and track the planes consistently across the entire sequence and an efficient multiple-view self-calibration algorithm based on the homographies induced by the scene plane. We show that by taking advantage of the presence of planar structures in the scene, our method avoids the difficulties of conventional SFM techniques in handling plane degeneracy and dynamic foreground, and hence highly complements those techniques in processing real-world commercial and consumer videos.

Chapter 3

Plane-Based Content-Preserving Warps for Video Stabilization

In Chapter 2, we proposed a reliable and efficient approach to structure and motion recovery by extracting the geometric information encoded in large scene planes. In fact, in addition to boosting the performance of SFM system, the recovered planar structures can be used to facilitate many other 3D modeling tasks as well. In this chapter, we will focus on the video stabilization problem, and investigate how the information about planar scene structures can be integrated into existing method to produce the state-of-the-art stabilization results.

3.1 Introduction

With the fast development of hand-held digital cameras, we have seen a dramatic increase in the amount of amateur videos shot over the past decade. However, very often people find their videos hard to watch, mainly due to the excessive amount of shake and undirected camera motions in the footage. Therefore, there has been an urgent demand in developing high-quality video stabilization algorithms, which are able to remove the undesirable jitters from amateur videos so that they appear to be taken under smooth, directed camera paths.

In general, there are two major steps in stabilizing a jittery input video, namely (1) designing new smooth camera paths, and (2) synthesizing stabilized video frames according to the new path. In this chapter, we focus on the second step, which still remains a highly challenging problem nowadays. Most existing methods [76, 47, 27, 61, 51] apply a full-frame 2D transformation to each input frame to obtain the stabilized output frame. Despite its computational efficiency and robustness, this approach is well-known for its inability in

handling the parallax effects of a non-degenarate scene and camera motion, as illustrated in Figure 3.1 (first row).

In fact, in the ideal case one will need the *dense* 3D structures of the scene in order to create a novel view of it. However, obtaining such a dense reconstruction from 2D images is extremely challenging in terms of both effectiveness and efficiency. Several attempts have been made along this direction [17, 40, 11], which rely on image-based rendering (IBR) to generate new images of a scene as seen along the smooth camera path. But these techniques are all limited to static scenes, among other issues. In a recent work [66], Liu et al. propose a novel method, namely content-preserving warping (CPW), which instead uses the *sparse* 3D points obtained by any structure from motion system for synthesis. The key idea of CPW is that the true dense deformation can be well approximated by diffusing the sparse displacements suggested by the reconstructed 3D points via a carefully chosen regularization term. This approximation is shown to be sufficient for stabilization, producing state-of-the-art results in many challenging cases, as long as there are enough feature tracks in each image region. In practice, however, large textureless regions often exist in the scene, such as ground, building facades, and indoor walls, where feature tracks are rare. It has been noticed that CPW performs poorly in these regions, as illustrated in Figure 3.1 (second row).

In this chapter, we propose a new synthesizing scheme which aims to remedy this important issue of CPW. Our key observation is that real scenes often exhibit strong structural regularities, in the form of *one or more planar surfaces*, which are largely ignored so far by existing methods. More importantly, these planar surfaces typically correspond to the textureless regions in the scene, which are problematic to CPW as well as many other methods.

Therefore, our goal is to develop a novel 3D stabilization method that can explicitly take advantage of the presence of (relatively large) planar surfaces in the scene. To this end, we propose to automatically detect large planes in the scene, and partition each frame into regions associated with each plane, as well as regions that are "non-planar". Note that, since our ultimate goal is to improve the stabilization system and produce jitter-free videos,

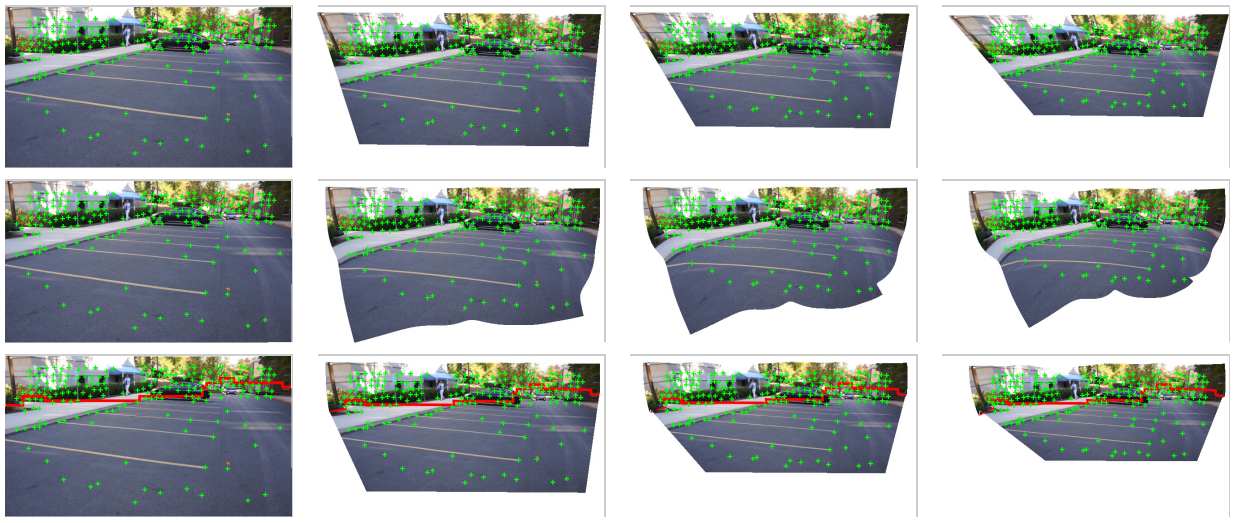


Figure 3.1: **Effects of various warping methods.** Each row shows a sequences of warps of a single input frame created by pulling the camera away from its original location. **First row:** Warping based on 2D transformation (e.g., homography) is too rigid to handle general motion and structures, resulting in large distortions in non-planar regions (e.g., buildings). **Second row:** Content-preserving warping preserves the non-planar structures well, but yields increasingly visible distortion in the textureless regions (i.e., the ground) where features are rare. **Third row:** Our plane-based warping is able to produce visually pleasing results by combining the strengths of both methods. Red line represents the boundary of planar and non-planar regions obtained by our video segmentation algorithm.

it is crucial for our segmentation algorithm to process the entire video in a short period of time, and obtain results which can be seamlessly integrated into the stabilization pipeline. To achieve this goal, we develop a novel algorithm which directly works on the same uniform grid mesh that is employed by CPW, and only uses geometric cues for fast processing. This is contrary to the existing piecewise planar scene segmentation algorithms, which operate at the per-pixel level and rely on multiple low-level and high-level photometric cues. These methods are generally too slow for stabilization purposes, taking hours to process a video with a few hundred frames. We demonstrate that our algorithm is capable of processing the entire video in about 30 seconds, and obtaining results that are sufficient for stabilization.

With the segmentation information, our new plane-based warping method computes a single homography for image regions that belong to the same plane, while borrowing the results of CPW for non-planar regions (Figure 3.1 third row). In this way, we not only

seamlessly integrate the information about planar structures of the scene into the stabilization framework, but also provide an unified framework for 2D-3D stabilization. When the scene is dominated by complex non-planar or dynamic structures, our method becomes CPW which is known to work well in such cases, whereas on the other end, if the scene contains a single large plane, it reduces to the robust and efficient 2D method.

3.1.1 Related Work

In general, depending on the level of scene geometry one recovers, existing video stabilization techniques can be roughly divided into two categories. Methods in the first category [76, 47, 27, 61, 51] aim to estimate a single 2D transformation between each pair of frames. Stabilization is then obtained by smoothing the parameters of 2D transformations followed by synthesizing a new video using the smoothed parameters. It is well known that 2D stabilization can only achieve limited smoothing before introducing noticeable artifacts to the output video. Several ideas have been examined in recent years to alleviate this problem, including interpolating the homography matrices in a transformed space [47], considering user's capturing intention [27], directly smoothing a set of robust feature trajectories [61], and designing an ℓ_1-optimal camera path [51].

In order to fully handle general scene structure and camera motion, 3D stabilization methods [17, 40, 11, 66] attempt to recover true camera motion and scene structures via structure from motion (SFM) systems. Stabilization is subsequently done by smoothing the camera path in 3D and synthesizing a new video based on the smoothed path. To avoid the dependency on structure from motion techniques, [67] directly smoothes the 2D feature trajectories based on the observation that they approximately lie in a low-dimensional subspace over any short period of time. Alternatively, [48] resorts to epipolar point transfer, which only requires projective reconstruction. However, all these methods except [48] solely rely on features that allow reliable tracking, and hence suffer from the presence of large textureless regions. In [48], epipolar constraints are used to search for additional matches

along edges. But this approach is very sensitive to noise, and does not work if there is no strong edge in the scene. Recently, [68] proposed to use additional depth sensors to compensate for the lack of feature tracks, but access to depth data is unrealistic for the vast majority of amateur videos.

The problem of segmenting video into motion layers that admit parametric transformation models was first studied in [112], and remains an active research topic in computer vision today. Since its goal is to obtain simultaneous motion estimation and segmentation, it typically involves iterative schemes which are prone to local minima. Given camera motion and 3D point cloud, early works on piecewise-planar scene segmentation from multiple images [4, 114] are based on line grouping and plane sweeping, whose complexity is prohibitive beyond a few images. More recently, [5] and [105] both combine the idea of random sampling consensus (RANSAC) with photometric consistency check to obtain piecewise planar scene models. However, the experiment results in both papers only involve simple examples with little non-planar structure. In addition, their computational complexity is still too high for our purpose. For example, it is reported in [105] that it takes 14 hours to process a sequence consisting of 380 frames. Finally, planes extracted from 3D point clouds or depth maps have been recently explored to improve the performance of multi-view stereo (MVS) systems [97, 43, 44, 80]. But these methods are again too slow for more than a few images. In summary, none of the existing methods meets our goal of obtaining satisfactory segmentation results within a few seconds for long video sequences.

3.2 Overview of the Content-Preserving Warping Technique

Since our method is built upon the content-preserving warping (CPW) technique introduced in [66], in this section we give a brief review of it.

Generally speaking, CPW is an image warping technique specifically designed for 3D sta-

bilization, which aims to deform an input frame according to a set of 2D sparse displacement constraints induced by the 3D viewpoint change, while minimizing the distortion of local shape and salient image content. In particular, it takes two sets of corresponding 2D points as input – $\hat{P}$ in the input frame, and P in the output frame – and create a dense warp guided by the displacements from $\hat{P}$ to P. For 3D stabilization, $\hat{P}$ and P are obtained by projecting the reconstructed 3D points into input and output (stabilized) cameras, respectively.

To create the dense warp, CPW first divides the original video frame $\hat{I}$ into an $m \times n$ uniform grid mesh, represented by a set of N vertices $\hat{V} = \{\hat{\boldsymbol{v}}_q\}_{q=1}^{N}$. Then, it estimates a warped version of the mesh, denoted by $V = \{\boldsymbol{v}_q\}_{q=1}^{N}$, for the output frame by minimizing the following objective function:

$$E(V) = E_d(V) + \alpha E_s(V), \tag{3.1}$$

where α is a scalar weight between the data term $E_d(V)$ and smoothness term $E_s(V)$.

Data term. The data term penalizes the difference in the output frame between the projected location of each point P_t and the location suggested by the estimated mesh V. For each point $\hat{P}_t$ in the input frame, a bilinear interpolation of the four corners of the enclosing grid cell, denoted by $\hat{V}_t$, is first computed so that $\hat{P}_t = w_t^T \hat{V}_t$. Here, the vector w_t contains the four coefficients that sum to 1. Then, the data term is defined as:

$$E_d(V) = \sum_t \|w_t^T V_t - P_t\|^2. \tag{3.2}$$

Smoothness Term. The smoothness term measures the deviation of the estimated 2D transformation of each grid cell from a *similarity transformation*. This is inspired by the work [56], which suggests that warps resembling a similarity transformation can effectively avoid noticeable distortions of image content due to shearing and non-uniform scaling, and hence should be preferred as long as the viewpoint change is not too large, which is indeed

the case in video stabilization. [56] further shows that this constraint can be written in the form of every three vertices that form a triangle in a grid cell. Specifically, let $(\hat{V}_1^\Delta, \hat{V}_2^\Delta, \hat{V}_3^\Delta)$ and $(V_1^\Delta, V_2^\Delta, V_3^\Delta)$ denote the vertices of any triangle Δ in the input and output grid mesh, respectively. Then, its deviation from a similarity transformation can be written as

$$\boldsymbol{e}_s(\Delta) = \|V_1^\Delta - (V_2^\Delta + a_\Delta(V_3^\Delta - V_2^\Delta) + b_\Delta R_{90}(V_3^\Delta - V_2^\Delta))\|^2, \tag{3.3}$$

where a_Δ, b_Δ satisfy

$$\hat{V}_1^\Delta = \hat{V}_2^\Delta + a_\Delta(\hat{V}_3^\Delta - \hat{V}_2^\Delta) + b_\Delta R_{90}(\hat{V}_3^\Delta - \hat{V}_2^\Delta), \tag{3.4}$$

and $R_{90} = [0\ 1; -1\ 0]$ is a 2D rotation matrix.

Finally, the smoothness term $E_s(V)$ is the sum of $\boldsymbol{e}_s(\Delta)$ over all eight triangles of each vertex:

$$E_s(V) = \sum_\Delta \boldsymbol{e}_s(\Delta). \tag{3.5}$$

Since minimizing the energy $E(V)$ is a linear least-squares problem in the set of unknown V, it can be solved efficiently by any standard linear system solver. The output frame is then generated using standard texture mapping algorithm according to V.

Finally we note that, according to the above discussion, the warp obtained by CPW tends to be close to a *similarity transformation*, especially in regions where features are rare or non-existing. However, similarity transformation cannot faithfully represent the projective effects of the scene, and hence may cause serious wobble effects in the stabilized videos. Next, we show how this problem can be properly addressed by incorporating information about scene planes.

3.3 Fast Piecewise Planar and Non-Planar Scene Segmentation for Videos

In this section, we propose a fast two-step approach to automatically segment each video frame into piecewise planar and non-planar regions. First, we detect scene planes from 3D point cloud obtained by structure from motion using a robust multiple structure estimation algorithm called J-Linkage [104]. Second, we describe a novel video segmentation algorithm, which classifies each grid cell in the CPW framework into $K + 1$ classes – one for each of the K detected planes, plus a "non-planar" class. For this problem, we lay out a MRF formulation for the entire sequence to simultaneously take into account the spatial coherence between neighboring cells within each frame, and improve the segmentation consistency across different frames. We now describe these two steps in detail.

3.3.1 Multiple Plane Detection

Since real scenes often contain multiple planes as well as non-planar structures, we adopt a robust multiple structure estimation method called J-Linkage [104] to detect planes from 3D point cloud. Similar to the popular RANSAC technique, this method is based on sampling consensus. Meanwhile, it has been shown in [104] that J-Linkage substantially outperforms other variants of RANSAC for multiple structure detection, such as sequential RANSAC and multi-RANSAC [122], in many real applications including 3D plane fitting.

Basically, J-Linkage works in the following way. It first generates a large number (typically a few thousands) of putative models by random sampling. Next, for each data point, a preference set (PS) of models is computed, which include all the models to which the distance from that data point is less than a threshold ϵ. J-Linkage then uses a bottom-up scheme to iteratively group data points that have similar PS. Here, the PS of a cluster is defined as the intersection of the preference sets of its members. Specifically, in each iteration, J-Linkage

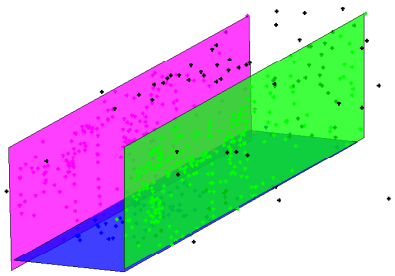


Figure 3.2: Three planes are detected by J-Linkage [104] on the video shown in Figure 3.3.

computes the Jaccard distance between any two clusters A and B:

$$d_J(A,B) = \frac{|A \bigcup B| - |A \bigcap B|}{|A \bigcup B|}, \tag{3.6}$$

and merge the two clusters with the smallest distance. As in RANSAC, the only free parameter of J-Linkage is the consensus threshold ϵ, which is set to 10 in our experiments. Also, since our goal is to detect large scene planes, we only keep those clusters with a support size larger than one sixth of the total number of points.

Figure 3.2 shows the result of applying J-Linkage to the 3D point cloud for an indoor video sequence taken by a person walking down the corridor with a hand-held camera (see Figure 3.3 for some input frames). In this example, three planes are detected, namely the ground and two side-walls. Although J-Linkage fails to detect the other two planes, namely the ceiling and front door, due to their small support sizes, we still consider the result successful as these two planes only occupy a very small portion of the video frames.

3.3.2 A Markov Random Field Formulation for Video Segmentation

Once a set of dominant planes is detected, the next step is to perform piecewise planar and non-planar segmentation for each input frame. To take both spatial and temporal consistency into consideration, we define a Markov random field for the entire sequence. For each frame, I_f, $f = 1, \ldots, F$, we divide it into a 64×36 uniform grid mesh and build a graph

Figure 3.3: **Piecewise planar and non-planar scene segmentation. Top Row:** Results of classifying each 3D point (represented by its image in each frame) into the $K+1$ classes based on the proposed distance measure $\|\boldsymbol{x}-\boldsymbol{x}_k^*\|_2$. Each color represents a class, with black circles corresponding to the points labeled as "non-planar", i.e., $\|\boldsymbol{x}-\boldsymbol{x}_k^*\|_2 > \beta, \forall k$. **Bottom Row:** Segmentation results obtained by the proposed method.

$\mathcal{G}_f = (\mathcal{V}_f, \mathcal{E}_f)$ on it. Each vertex $p \in \mathcal{V}_f$ is a cell of the mesh, while the edges, $\mathcal{E}_f$, denote the neighboring relationship between cells. Then, the graphs $\{\mathcal{G}_f\}$ from all frames are merged into a large graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, by adding edges between the two cells at the same spatial location in two consecutive frames.

Given a set of K 3D planes, our goal is to assign a unique label l_i to each vertex $p_i \in \mathcal{V}$. That is, $l_i = k, k = 1, 2, \ldots, K$ if p_i belongs to the k-th plane, and $l_i = 0$ if p_i lies on any non-planar surface. The solution $L = \{l_i\}$ can be obtained by minimizing the energy function

$$E(L) = \sum_{p_i \in \mathcal{V}} \Psi_i(l_i) + \sum_{e_{ij} \in \mathcal{E}} \Psi_{ij}(l_i, l_j), \tag{3.7}$$

which involves a unary data function Ψ_i and a pairwise smoothness function Ψ_{ij}. In this chapter, we adopt the popular multi-label graph-cut algorithm [14] to minimize $E(L)$. It is guaranteed to find a solution that is within a constant factor of the global minimum, and has been shown to produce satisfactory results in many vision tasks [102].

Data term. For a vertex in the f-th frame, $p_i \in \mathcal{V}_f$, the function Ψ_i is defined as follows. Let $\mathcal{X}_i$ be the set of 3D points whose images in the f-th frame lie in the cell corresponding to p_i. Then, for each point $X \in \mathcal{X}_i$, we compute its projection to the k-th plane, denoted as X_k^*. We further denote $\boldsymbol{x}$ and $\boldsymbol{x}_k^*$ as the images of X and X_k^* in the f-th frame, respectively.

The function Ψ_i then measures the image distance between $\boldsymbol{x}$ and $\boldsymbol{x}_k^*$:

$$\Psi_i(l_i) = \begin{cases} \sum_{X \in \mathcal{X}_i} \min\{\|\boldsymbol{x} - \boldsymbol{x}_k^*\|_2, d_{\max}\}, & \text{if } l_i = k > 0 \\ \beta|\mathcal{X}_i|, & \text{if } l_i = 0 \end{cases} \tag{3.8}$$

where β is a penalty assigned to each point $X \in \mathcal{X}_i$ if the corresponding cell is classified as "non-planar". Note that, geometrically, β can be viewed as a threshold that determines how far the images of X and its projection onto the k-th plane X_k^* may be before X is considered not belonging to that plane. On one hand, by comparing the image distance instead of the distance in 3D, β sets a uniform threshold across all 3D points which is irrelevant to their individual uncertainty in the 3D space. On the other hand, the value of β should depend on the overall accuracy of structure from motion, and is chosen to be 1.5 times the size of each cell in our work. For example, for a 640×360 input frame, we have $\beta = 15$. In addition, the distance measure has been truncated in Eq. (3.8) to $d_{\max}$ in order to prevent it from being dominated by a small number of poorly reconstructed 3D points. We fix $d_{\max} = 2\beta$ for all the experiments.

In Figure 3.3 (first row) we show the results of classifying each 3D point (represented by its image in each frame) into the $K + 1$ classes based on the proposed distance measure for an indoor scene. As one can see, the classification results indeed give us very strong cues for segmentation.

Smoothness term. For each edge $e_{ij} \in \mathcal{E}$ in the same image I_f, the smoothness function is defined as:

$$\Psi_{ij}(l_i, l_j) = \delta(l_i, l_j) \cdot g(i, j), \tag{3.9}$$

where $\delta(l_i, l_j)$ is the indicator function which takes value 0 if $l_i = l_j$, and 1 otherwise.

The function $g(i, j)$ is designed to improve the estimation of label boundaries by imposing geometric constraints derived from multiple planes in the scene. First, for each pair of planes in the scene (if one exists), we compute the 2D intersection line L between them in each

frame I_f. Then, we find all pairs of neighboring cells (p_i, p_j) in I_f where the centers of p_i and p_j lie on different sides of L, and accumulate all such pairs for all intersection lines in a set $\mathcal{E}_f^L$. Finally, the function $g(i,j)$ is defined as

$$g(i,j) = \begin{cases} \lambda_1, & \text{if } (p_i, p_j) \notin \mathcal{E}_f^L \\ \lambda_2, & \text{otherwise} \end{cases} \tag{3.10}$$

For edges e_{ij} across two frames, the smooth cost is defined as

$$\Psi_{ij}(l_i, l_j) = \lambda_3 \delta(l_i, l_j). \tag{3.11}$$

In this work, λ_1, λ_2 and λ_3 are empirically set to $\lambda_1 = \lambda_3 = 10$, $\lambda_2 = 2$ for all experiments.

In Figure 3.3 and Figure 3.4, we show some representative results of the proposed method. As one can see, our segmentation algorithm correctly identifies the large planar regions in a variety of indoor and outdoor scenes. However, since our algorithm purely relies on geometric cues, the label boundaries estimated by it may not be very accurate. This is mainly due to the uncertainty in 3D reconstruction, which decides the smallest possible threshold β one can choose to distinguish points on a plane from others. In addition, the facts that our algorithm only operates on a coarse spatial grid, and that feature points are not evenly distributed in the images, could also contribute to the errors. Nevertheless, we find that these errors have little effect on the final stabilization results, since the shifts in viewpoint are usually small for video stabilization.

In terms of speed, for a typical sequence such as the one shown in Figure 3.3 with 250 frames, the plane detection[1] and piecewise planar scene segmentation algorithms take about 10 and 15 seconds on a desktop PC with 3.40GHz CPU and 12GB memory, respectively.

[1]We use the Matlab code from J-Linkage website: http://www.diegm.uniud.it/fusiello/demo/jlk/.

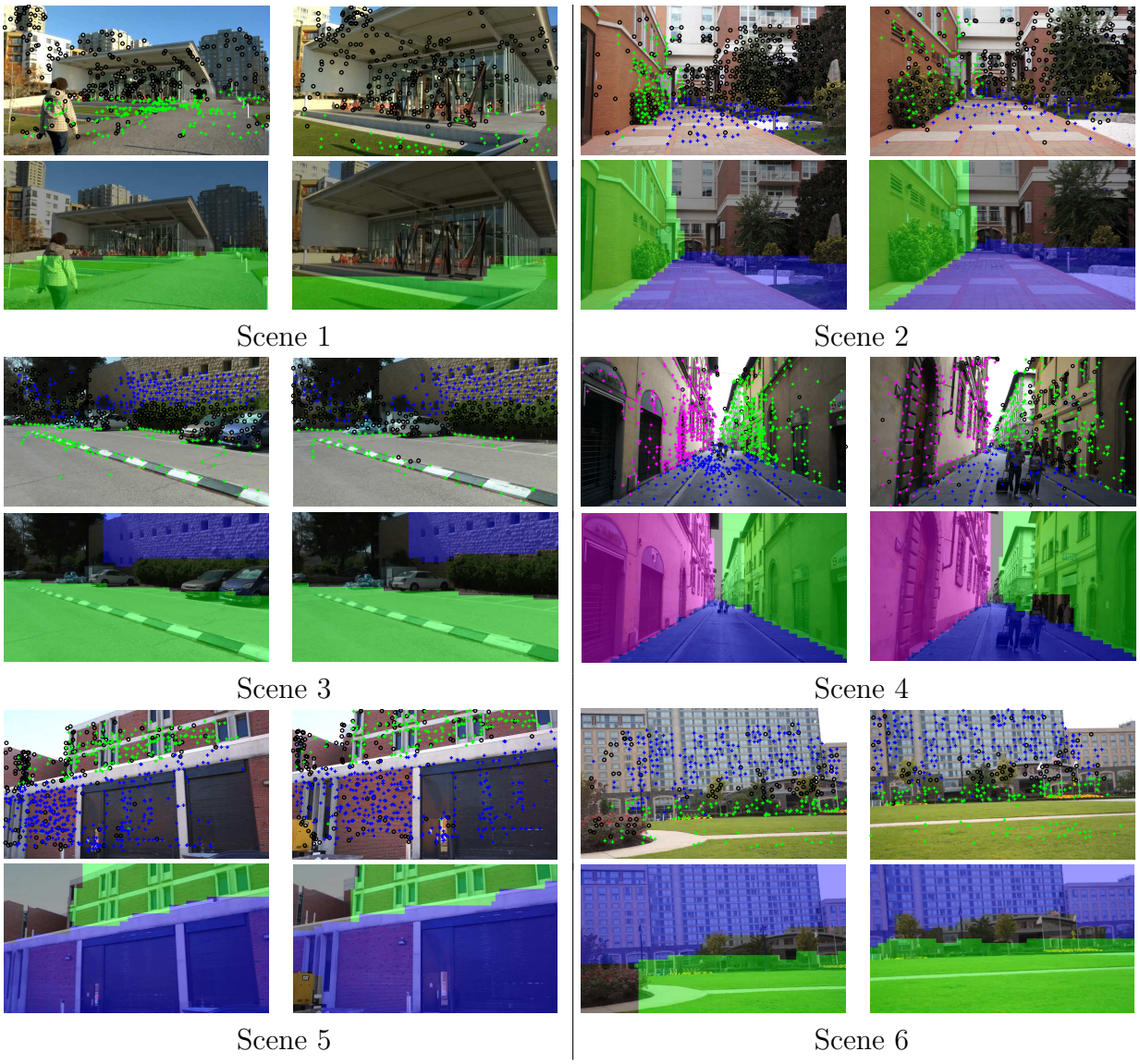


Figure 3.4: Additional results on piecewise planar and non-planar scene segmentation.

3.4 Plane-Based Stabilization

As we have already discussed, this chapter aims at leveraging the flexibility of CPW and the structural regularities (i.e., planar surfaces) of the scene to produce high-quality stabilization results, especially in the cases where CPW performs poorly because of large textureless regions. In this section, we describe our plane-based stabilization algorithm in detail.

Like other 3D stabilization methods, our plane-based method first applies structure from

motion to recover the original camera motion and sparse 3D point cloud. Here, we use ACTS [117], a publicly available structure from motion system. To generate the stabilized camera path, we apply Gaussian filter to the original camera parameters. Since a camera can be modeled by a rotation matrix $R \in SO(3)$ and its center $C \in \mathbb{R}^3$, we apply a Gaussian filter to these two components separately. Note that, since the space of rotation matrices is not Euclidean, the filtering of the rotational component is done in a locally linearized space at each timestamp in the same way described in [66].

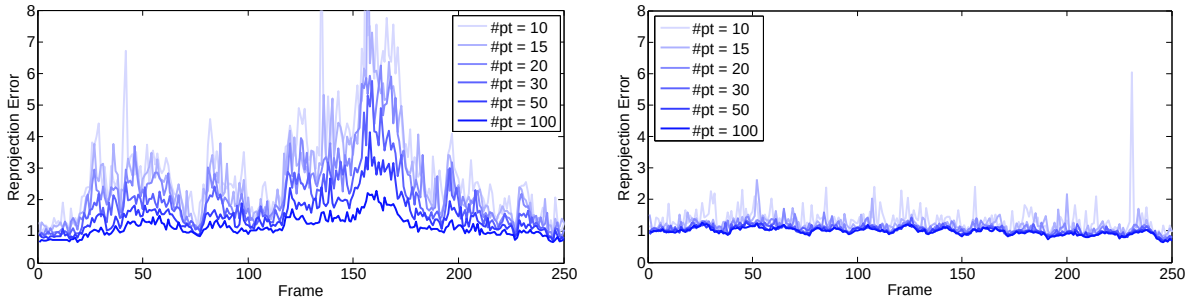
For novel view synthesis, we also follow the same idea of [66] by processing one input frame at a time to avoid ghosting effect caused by the moving objects. Each input frame is divided into a 64×36 grid mesh $\hat{V} = \{\hat{\boldsymbol{v}}_q\}_{q=1}^{N}$ and the content-preserving warp is then computed. We denote the output mesh by $V^0 = \{\boldsymbol{v}_q^0\}$. To incorporate information about the piecewise planar scene structures into stabilization, we give a label, l_q, to each vertex of the mesh according to the labels of its surrounding cells. For any vertex that lies on the segmentation boundary (hence the surrounding cells have more than one labels), we simply assign the smallest label to it. Based on the labels, a new mesh $V = \{\boldsymbol{v}_q\}$ is computed:

$$\boldsymbol{v}_q = \begin{cases} H_k \hat{\boldsymbol{v}}_q & \text{if } l_q = k, k = 1, \ldots, K \\ \boldsymbol{v}_q^0, & \text{if } l_q = 0 \end{cases} \tag{3.12}$$

where H_k is the homography induced by the k-th plane between the input and output frames. The output frame is then obtained using standard texture mapping algorithms.

3.4.1 Quantitative Comparison of Two Warping Methods

In the rest of this section, we provide a quantitative performance comparison of our plane-based warping with CPW in handling planar regions in the scene. In particular, we are interested in evaluating the performances of both methods with respect to the number of available feature tracks. Therefore, we choose the video shown in the upper-left corner of



(a) CPW (b) Our plane-based warping

Figure 3.5: Average reprojection errors (pixels) achieved by both methods with various number of feature tracks.

Figure 3.6 for this experiment, in which the scene is dominated by a large plane (the ground) and the tracking algorithm is able to obtain a large number of feature tracks (> 200) for each frame.

Given the estimated 3D structure and camera motion, we randomly select m_p feature tracks for each input frame, and use them to compute the warps between the input and output frames using both warping methods. Then, we use the rest of the feature tracks to evaluate the accuracy of both warping methods. Specifically, let $\hat{P}_t$ and P_t denote the locations of a feature point in input and output frames, respectively, and $\tilde{P}_t$ be the estimated location of this feature point in the output frame given $\hat{P}_t$ using either warping method. We compute the average reprojection error for each frame as follows:

$$err(f) = \frac{1}{|\mathcal{P}_f|} \sum_{t \in \mathcal{P}_f} \|\tilde{P}_t - P_t\| \tag{3.13}$$

where $\mathcal{P}_f$ is the set of test feature points in frame f.

In Figure 3.5 we show the average reprojection error $err(f)$ of both methods with various numbers of feature tracks m_p. As expected, The reprojection error of CPW increases significantly as m_p decreases. On the contrary, the performance of our plane-based method remains stable with a small number of features, justifying its advantage in handling texture-

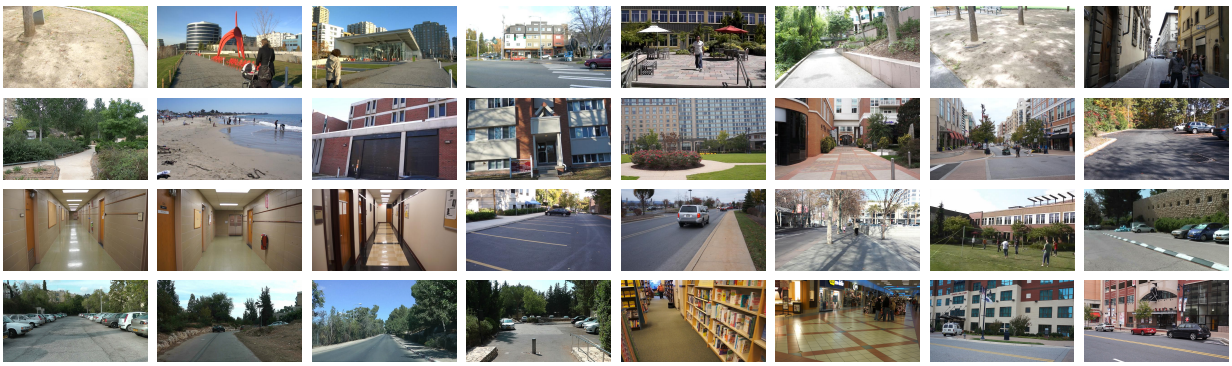


Figure 3.6: Snapshots of the videos used for evaluation.

less regions in the scene.[2] Meanwhile, it also worth noting that the performance of CPW varies from frame to frame. The reason is at least two-fold. First, the motion between the input and output frames is different for each frame. Second, CPW is more sensitive to certain types of motion (e.g., camera rotations) than others (e.g., translations) as it uses a similarity transformation to regularize the warping.

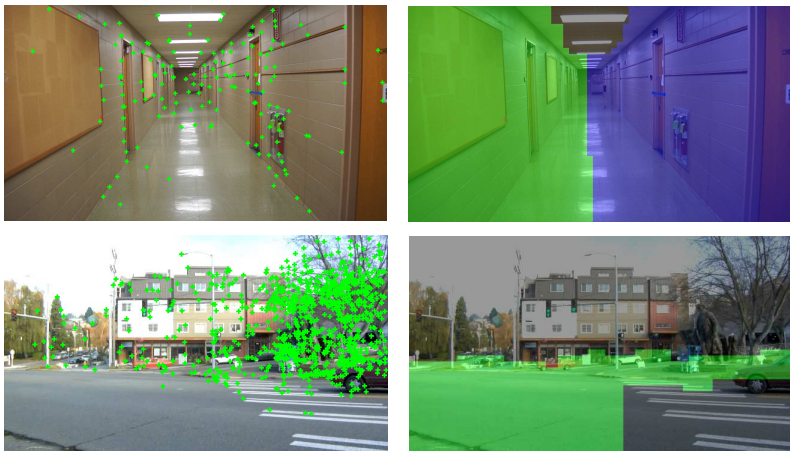
3.5 Video Stabilization Results

We have tested our algorithm on 32 video sequences (see Figure 3.6) which consist of one or more large scene planes,[3] including 5 videos that are used in [66] to demonstrate the performance of CPW. These sequences cover a wide range of scenes from both natural and indoor/outdoor man-made environments. Among them, noticeable wobble effects can be seen in 18 results obtained by CPW, due to the lack of feature tracks in large planar regions. Meanwhile, our plane-based method succeeds in 30 of the 32 videos, generating satisfactory stabilization results.

Challenging cases. For the other two testing videos shown in Figure 3.7, our method is not able to completely remove the wobble effects, although it still produces better results

[2]In theory, four reliable feature tracks are sufficient to compute the plane-based warping (i.e., homography) for each plane in the scene. In practice, however, a slightly large number may be necessary due to the presence of noise.

[3]More precisely, we mean that J-Linkage detects at least one plane in each of these videos. For videos with no plane detected, our method simply becomes CPW so no comparison needs to be done.



(a) Input frame with points (b) Segmentation result

Figure 3.7: **Challenging cases for our method. Top row:** In this case, only a very small number of points are detected on the ground. Some of them actually correspond to the reflection. **Bottom row:** In this case, the ground is slightly curved.

than CPW. In the first video, only a very small number of points are reconstructed on the ground, with a large number of outliers due to reflection. Therefore, J-Linkage fails to detect the ground plane in the case. Consequently, our segmentation algorithm incorrectly assigns the ground regions to the planes corresponding to the walls, causing undesirable artifacts in the stabilized video. In the second video, the ground is slightly curved, which confuses our plane detection and segmentation algorithms. As a result, a portion of the ground region is labeled as non-planar, hence the wobble effects remain in the output video.

In fact, both cases reveal the dependency of our method's performance on a few free parameters in the plane detection and segmentation algorithms, for which a set of fixed values is certainly not enough to handle all cases. Nevertheless, we have shown in this chapter that, by exploiting information about scene structures such as the planar surfaces, our method significantly outperforms CPW in many challenging cases.

3.6 Conclusion, Limitations, and Future Work

In this chapter we have described a novel method for video stabilization, which outperforms the state-of-the-art methods by taking advantage of the presence of large planes in the scene. Our method is built upon the newly proposed CPW framework, but is able to avoid the difficulties of CPW in handling large textureless regions. In particular, we have proposed an efficient Markov random field formulation to segment each video frame into piecewise planar and non-planar regions. This level of scene understanding is shown to be ideal for generating high-quality jitter-free videos in a variety of practical scenarios.

Like CPW and many other 3D methods, our algorithm relies on structure from motion to get accurate information about the 3D scene structures and camera motions. For this reason, all the videos tested in this work are chosen to be friendly to SFM. Also, we do not address other common issues in video stabilization, including the smaller field of view, motion blur [76], and rolling shuttle effects [50].

Another bottleneck of our method is the plane detection part. Currently we use the robust model estimation package J-Linkage, but it leaves to the user to decide the minimum number of inliers for a valid model; hence it may fail when the number of reconstructed 3D points on the plane is extremely small. A different direction would be combining plane detection with 3D reconstruction, as we discussed in Chapter 2.

Chapter 4

Low-Rank Matrix Recovery via Convex Optimization

In the previous two chapters, we systematically studied the problem of recovering 3D planar structures and camera motions from 2D image sequences, and demonstrated the application of the recovered planar structures in stabilizing shaky amateur videos. In the next three chapters, we move away from planar structures and investigate other types of regularities in the visual data, including symmetric or regular patterns in an image and the linear correlations among multiple images, which can be captured by a low-rank structure model. In this chapter, we introduce a new method for recovering low-rank matrices from corrupted observations via convex optimization, with a focus on the stability of this new method when applied to noisy data. We show how this method can be extended and employed to solve 3D reconstruction problems in Chapter 5 and 6.

4.1 Introduction to Principal Component Pursuit

The advance of modern information technologies has produced a tremendous amount of high-dimensional data in science, engineering, and society, such as images, videos, web documents, and bioinformatics data. It has become a pressing challenge to develop efficient and effective tools to process, analyze, and extract useful information from such high-dimensional data. One of the fundamental problems here is how to extract the intrinsic low-dimensional structure of such high-dimensional data.

Arguably, the classical *Principal Component Analysis* (PCA) [34, 58] is the most widely used statistical tool for high-dimensional data analysis and dimensionality reduction today.

It basically assumes that the data approximately lie on a low-dimensional linear subspace. Mathematically, if we stack all the data points as column vectors of a matrix M, then the matrix should be approximately low-rank and can be written as $M = L_0 + Z_0$, where L_0 is a low-rank matrix (representing the subspace) and Z_0 models a small noisy perturbation of each entry of L_0. Then, PCA simply seeks the best rank-k estimate of L_0 in the ℓ_2 sense, which can be solved efficiently via singular value decomposition (SVD) and thresholding. It can be shown that if the perturbation is i.i.d. Gaussian, this gives a statistically optimal estimate of the subspace. Such an estimate is naturally stable in the sense that the error is bounded to be proportional to the magnitude of the perturbation.

However, it is well known that the classical PCA breaks down even with a single grossly corrupted entry in the data matrix M, i.e., it is *not robust* to gross errors or outliers. Many methods have been proposed to alleviate this problem; however, none of them yield a polynomial-time algorithm with strong performance guarantees (see [18] for a detailed discussion). Unlike the previous methods, the recently proposed Principal Component Pursuit (PCP) method utilizes a convex program that guarantees to recover a low-rank matrix despite gross sparse errors under rather broad conditions. In the rest of this section, we give a brief review of this method.

Mathematically, PCP considers the matrix M of the form $M = L_0 + S_0$, where L_0 is low-rank and S_0 is a sparse matrix with most of its entries being zero. Unlike the model for PCA, here both components can be of arbitrary magnitude and no other information about the rank of L_0 and/or the support or signs of S_0 is given. To recover L_0 and S_0, PCP solves the following convex optimization problem[1]

$$\min_{L,S} \ \|L\|_* + \lambda\|S\|_1 \quad \text{subject to} \quad M = L + S. \tag{4.1}$$

[1]In this chapter, we use five norms of a matrix A. $\|A\|_*$ denotes its nuclear norm – sum of its singular values, $\|A\|_F$ denotes its Frobenius norm and $\|A\|$ denotes its 2-norm. Moreover, $\|A\|_1$ and $\|A\|_\infty$ are the ℓ_1 and ℓ_∞ norms of A viewed as a vector, respectively.

It has been shown in [18] that, under surprisingly broad conditions, the above convex program exactly recovers L_0 and S_0.[2] Specifically, let $L_0 = U\Sigma V^* = \sum_{i=1}^{r} \sigma_i u_i v_i^*$ denote the singular value decomposition of $L_0 \in \mathbb{R}^{n_1 \times n_2}$, where r is the rank, $\sigma_1, \ldots, \sigma_r$ are the singular values, and $U = [u_1, \ldots, u_r], V = [v_1, \ldots, v_r]$ are the matrices of left- and right-singular vectors, respectively. The incoherence conditions on U and V with parameter μ are as follows:

$$\max_i \|U^* e_i\|^2 \leq \frac{\mu r}{n_1}, \ \max_i \|V^* e_i\|^2 \leq \frac{\mu r}{n_2}, \ \|UV^*\|_\infty \leq \sqrt{\frac{\mu r}{n_1 n_2}}, \tag{4.2}$$

where e_i's are the canonical basis vectors. Now let $\|S_0\|_0 = m$ be the number of nonzero entries in S_0. The conditions on S_0 concern the identifiability issue arises when S_0 is also low-rank. To avoid such pathological cases, [18] assumes that the support of sparse component S_0 is selected uniformly at random among all subsets of size m. Under these conditions, the main result of [18] states:

Theorem 1 ([18]). *Suppose $L_0 \in \mathbb{R}^{n \times n}$ obeys* (4.2) *and that the support set of S_0 is uniformly distributed. Then there is a numerical constant c such that with probability at least $1 - cn^{-10}$ (over the choice of support of S_0), Principal Component Pursuit* (4.1) *with $\lambda = 1/\sqrt{n}$ recovers L_0 and S_0 exactly, provided that*

$$\operatorname{rank}(L_0) \leq \rho_r n \mu^{-1} (\log n)^{-2} \quad \text{and} \quad m \leq \rho_s n^2, \tag{4.3}$$

where ρ_r and ρ_s are some positive constants.

Note that the analysis and result of PCP apply to any rectangular $(n_1 \times n_2)$ matrix. But to simplify presentation, we have assumed that the matrices are all square and write $n = n_1 = n_2$. The modification needed for general rectangular matrices is straightforward and will be briefly discussed in the end of the chapter.

It is also worth noting that a large number of first-order techniques have been developed to

[2] Readers are also referred to [26] which proposed to solve the same problem but with different exact recovery conditions.

solve the convex program (4.1). In particular, we have found that the Augmented Lagrange Multiplier (ALM) method [103] works well in all the 3D reconstruction applications we study in this thesis. Interested readers are referred to [103] for detailed discussion of the method.

4.2 Stable Principal Component Pursuit

The PCP result [18] is limited to the low-rank component being exactly low-rank and the sparse component being exactly sparse. However, in real world applications such as 3D reconstruction the observations are often corrupted by noise, which may be stochastic or deterministic, affecting every entry of the data matrix. For example, in face recognition, the human face is not a strictly convex and Lambertian surface, so small perturbation accounting for the fact that the low-rank component is only approximately low-rank needs to be considered. In ranking and collaborative filtering, user's ratings could be noisy because of the lack of control in the data collection process. Therefore, for the techniques developed in [18] to be widely applicable, results that guarantee stable and accurate recovery in the presence of entry-wise noise must be established.

4.2.1 Assumption and Main Result

The new measurement model that we consider here assumes that we observe

$$M = L_0 + S_0 + Z_0, \tag{4.4}$$

where Z_0 is a noise term – say i.i.d. noise on each entry of the matrix. However, all we assume about Z_0 is that $\|Z_0\|_F \leq \delta$ for some $\delta > 0$. To recover the unknown matrices L_0 and S_0, we propose solving the following optimization problem, as a relaxed version to PCP (4.1):

$$\min_{L,S} \ \|L\|_* + \lambda \|S\|_1 \quad \text{subject to} \quad \|M - L - S\|_F \leq \delta. \tag{4.5}$$

where we choose $\lambda = 1/\sqrt{n}$. Our main result is that under the same conditions as PCP, the above convex program gives a stable estimate of L_0 and S_0:[3]

Theorem 2. *Suppose again that L_0 obeys (4.2) and the support of S_0 is uniformly distributed. Then if L_0 and S_0 satisfy (4.3) with $\rho_r, \rho_s > 0$ being sufficiently small numerical constants, with high probability in the support of S_0, for any Z_0 with $\|Z_0\|_F \leq \delta$, the solution $(\hat{L}, \hat{S})$ to the convex program (4.5) satisfies*

$$\|\hat{L} - L_0\|_F^2 + \|\hat{S} - S_0\|_F^2 \leq Cn^2\delta^2, \tag{4.6}$$

where C is a numerical constant.

The precise form of the constant C will be given in Proposition 4. Here, we would like to point out two ways to view the significance of this result. To some extent, our model unifies the classical PCA and the robust PCA by considering both gross sparse errors and small entry-wise noise in the measurements. So on one hand, our result says that the low-rank and sparse decomposition via PCP is stable in the presence of small entry-wise noise, hence making PCP more widely applicable to practical problems where the low-rank structure is not exact. On the other hand, together with the result of PCP [18], our new result convincingly justifies that the classical PCA can now be made robust to sparse gross corruptions via certain convex programs. Since this convex program can be solved very efficiently [65], at a cost not so much higher than the classical PCA, our result is expected to have significant impact on many practical problems.

4.2.2 Relations to Existing Work

Aside from its close relations to the classical PCA and the newly proposed robust PCA work mentioned above, our analysis and result are closely related to two lines of development,

[3]It can be shown that our result also implies stability in estimation of the singular values and singular subspace of L_0; see, for example, [100].

regarding stable recovery of sparse signals and low-rank matrices, respectively.

Conceptually, our work is very similar to the development of results for the "imperfect" scenarios in compressive sensing where the measurements are noisy and the signal is not exact sparse. More precisely, ℓ_1-norm minimization techniques are adapted to recover a vector $x_0 \in \mathbb{R}^m$ from contaminated observations $y = Ax_0 + z$ where $A \in \mathbb{R}^{n \times m}$ with $n \ll m$ and z is the noise term. After the landmark work of [23] which established that for the noise-free case, the minimal ℓ_1-norm solution exactly recovers the sparse signal under fairly broad conditions, later works have demonstrated that stable recovery occurs for most measurement ensembles [32], or particularly, when the measurement ensembles satisfy some incoherence conditions [33] or restricted isometry property (RIP) [22].

Recently, there has been an explosion of literature regarding the power of nuclear-norm minimization in recovering low-rank matrices from under-sampled measurements. A matrix RIP is first proposed by [92] to connect compressive sensing with low-rank matrix recovery. For measurement ensembles obeying the RIP, tight bounds of the recovery error from noisy data have been developed in [19] which is within a constant of the minimax risk and an oracle error. Also see [82] for similar results. Technically, our work is more closely related to the recent work [20] which developed the first stability result for the matrix completion problem under small perturbations. Naturally, in establishing the stability result for robust PCA, we borrow heavily from the techniques used in [20] and [18].

4.2.3 Notation and Outline of Analysis

Our goal is to show that in cases where the noise free PCP (4.1) *exactly* recovers (L_0, S_0), the noise aware version (4.5) *stably* estimates (L_0, S_0). In the noise free case, exact recovery is guaranteed by the existence of a "dual certificate" W described in Lemma 3 below. The main result of [18] is to show that under the stated conditions, with high probability such a dual certificate exists. Then Proposition 4 below shows that the existence of such a certificate also implies that the recovery via (4.5) under noise is stable.

Before continuing, we fix some notation. Given a matrix pair $X_0 = (L_0, S_0)$, let $\Omega \subseteq [n] \times [n]$ denote the support of S_0, and $\mathcal{P}_\Omega$ denote the projection operator onto the space of matrices supported on Ω. Let $r = \mathrm{rank}(L_0)$, and let $L_0 = U\Sigma V^*$ denote the compact SVD of L_0, with $U, V \in \mathbb{R}^{n\times r}$ and $\Sigma \in \mathbb{R}^{r\times r}$. We will let T denote the subspace generated by matrices with the same column space or row space as L_0:

$$T = \{UQ^* + RV^* \mid Q, R \in \mathbb{R}^{n\times r}\} \subset \mathbb{R}^{n\times n},$$

and $\mathcal{P}_T$ be the projection operator onto this subspace.

For any pair $X = (L, S)$ let $\|X\|_F \doteq (\|L\|_F^2 + \|S\|_F^2)^{1/2}$, and define the projection operator $\mathcal{P}_T \times \mathcal{P}_\Omega : (L, S) \mapsto (\mathcal{P}_T L, \mathcal{P}_\Omega S)$. Define the subspaces $\Gamma \doteq \{(Q, Q) \mid Q \in \mathbb{R}^{n\times n}\}$ and $\Gamma^\perp \doteq \{(Q, -Q) \mid Q \in \mathbb{R}^{n\times n}\}$, and let $\mathcal{P}_\Gamma$ and $\mathcal{P}_{\Gamma^\perp}$ denote their respective projection operators. Finally, for any linear operator $\mathcal{A} : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$, we use $\|\mathcal{A}\|$ to denote the operator norm $\sup_{\|X\|_F=1} \|\mathcal{A}X\|_F$.

With these notations, the optimality conditions for (4.1) can be stated in terms of a dual vector as follows.

Lemma 3 (Lemma 2.5 in [18])**.** *Assume that* $\|\mathcal{P}_\Omega \mathcal{P}_T\| \le 1/2$ *and* $\lambda < 1$*. Suppose that there exists* W *such that*

$$\begin{cases} W \in T^\perp, \quad \|W\| < 1/2, \\ \|\mathcal{P}_\Omega(UV^* - \lambda \mathrm{sgn}(S_0) + W)\|_F \le \lambda/4, \\ \|\mathcal{P}_{\Omega^\perp}(UV^* + W)\|_\infty < \lambda/2. \end{cases} \tag{4.7}$$

Then the pair (L_0, S_0) *is the unique optimal solution to* (4.1)*.*

From now on, we will write $\lambda \mathcal{P}_\Omega D = \mathcal{P}_\Omega(UV^* - \lambda \mathrm{sgn}(S_0) + W)$. The following proposition shows that under the existence of such a dual certificate, (4.5) will also stably recover L_0 and S_0 in the presence of noise.

Proposition 4. *Assume* $\|\mathcal{P}_\Omega \mathcal{P}_T\| \le 1/2$*,* $\lambda \le 1/2$*, and that there exists a dual certificate* W

satisfying (4.7). *Let* $\hat{X} = (\hat{L}, \hat{S})$ *be the solution to* (4.5) *and* $X_0 = (L_0, S_0)$, *then* $\hat{X}$ *satisfies*

$$\|X_0 - \hat{X}\|_F \leq (8\sqrt{5}n + \sqrt{2})\delta. \tag{4.8}$$

Proposition 4 implies Theorem 2, since under the conditions of Theorem 2, Lemma 2.8 and Lemma 2.9 of [18] show that with high probability, there indeed exists such a dual certificate W, and Corollary 2.7 of [18] proves $\|\mathcal{P}_\Omega \mathcal{P}_T\| \leq 1/2$ as well.

The rest of the section then sets out to prove Proposition 4 and is organized as follows. In Section 4.2.4, we prove two key lemmas on which our main result depends. The proof of Proposition 4 then follows in Section 4.2.5. We further provide numerical results in Section 4.2.6 to support our analysis and conclude the section with additional discussions in Section 4.2.7.

4.2.4 Two Lemmas

In this section, we prove two lemmas which will be useful in the development of our main result. For any matrix pair $X = (L, S)$, we define $\|X\|_\diamond = \|L\|_* + \lambda\|S\|_1$.

Lemma 5. *Assume* $\|\mathcal{P}_\Omega \mathcal{P}_T\| \leq 1/2$ *and* $\lambda \leq 1/2$. *Suppose that there exists a dual certificate* W *satisfying* (4.7) *and write* $\Lambda = UV^* + W$. *Then for any perturbation* $H = (H_L, H_S)$ *obeying* $H_L + H_S = 0$,

$$\begin{aligned} \|X_0 + H\|_\diamond \;\; \geq \;\; & \|X_0\|_\diamond + (3/4 - \|\mathcal{P}_{T^\perp}(\Lambda)\|)\|\mathcal{P}_{T^\perp}(H_L)\|_* \\ & + (3\lambda/4 - \|\mathcal{P}_{\Omega^\perp}(\Lambda)\|_\infty)\|\mathcal{P}_{\Omega^\perp}(H_S)\|_1. \end{aligned}$$

Proof. For any $Z = (Z_L, Z_S) \in \partial\|X_0\|_\diamond$, we have

$$\|X_0 + H\|_\diamond \geq \|X_0\|_\diamond + \langle Z_L, H_L\rangle + \langle Z_S, H_S\rangle.$$

Now due to the form of the subgradients of the ℓ_1 norm and the nuclear norm,[4] we have the identities: $Z_L = \Lambda + \mathcal{P}_{T^\perp}(Z_L - \Lambda)$ and $Z_S = \Lambda - \lambda\mathcal{P}_\Omega D + \mathcal{P}_{\Omega^\perp}(Z_S - \Lambda)$. Thus we have:

$$\begin{aligned}\langle Z_L, H_L\rangle + \langle Z_S, H_S\rangle &= \langle \Lambda, H_L\rangle + \langle \mathcal{P}_{T^\perp}(Z_L - \Lambda), H_L\rangle \\ &\quad + \langle \Lambda - \lambda\mathcal{P}_\Omega D, H_S\rangle + \langle \mathcal{P}_{\Omega^\perp}(Z_S - \Lambda), H_S\rangle \\ &\geq \langle Z_L - \Lambda, \mathcal{P}_{T^\perp}(H_L)\rangle + \langle Z_S - \Lambda, \mathcal{P}_{\Omega^\perp}(H_S)\rangle - \frac{\lambda}{4}\|\mathcal{P}_\Omega(H_S)\|_F\end{aligned}$$

since $H_L + H_S = 0$ and $\|\mathcal{P}_\Omega D\|_F \leq 1/4$.

Moreover, by duality, there exists $Z_L^* \in \partial\|L_0\|_*$ with $\|Z_L^*\| \leq 1$ such that $\langle Z_L^*, \mathcal{P}_{T^\perp}(H_L)\rangle = \|\mathcal{P}_{T^\perp}(H_L)\|_*$. Also notice that $|\langle \Lambda, \mathcal{P}_{T^\perp}(H_L)\rangle| = |\langle \mathcal{P}_{T^\perp}(\Lambda), \mathcal{P}_{T^\perp}(H_L)\rangle| \leq \|\mathcal{P}_{T^\perp}(\Lambda)\|\|\mathcal{P}_{T^\perp}(H_L)\|_*$. Therefore, let $Z_L = Z_L^*$, we have:

$$\langle Z_L - \Lambda, \mathcal{P}_{T^\perp}(H_L)\rangle \geq (1 - \|\mathcal{P}_{T^\perp}(\Lambda)\|)\|\mathcal{P}_{T^\perp}(H_L)\|_*.$$

Similarly, by duality, there exists $Z_S^* \in \partial(\lambda\|S_0\|_1)$ with $\|Z_S^*\|_\infty \leq \lambda$ such that $\langle Z_S^*, \mathcal{P}_{\Omega^\perp}(H_S)\rangle = \lambda\|\mathcal{P}_{\Omega^\perp}(H_S)\|_1$. Therefore, choose Z_S to be $Z_S = Z_S^*$, we have:

$$\langle Z_S - \Lambda, \mathcal{P}_{\Omega^\perp}(H_S)\rangle \geq (\lambda - \|\mathcal{P}_{\Omega^\perp}(\Lambda)\|_\infty)\|\mathcal{P}_{\Omega^\perp}(H_S)\|_1.$$

Observe now that

$$\begin{aligned}\|\mathcal{P}_\Omega(H_S)\|_F &\leq \|\mathcal{P}_\Omega\mathcal{P}_T(H_S)\|_F + \|\mathcal{P}_\Omega\mathcal{P}_{T^\perp}(H_S)\|_F \leq \frac{1}{2}\|H_S\|_F + \|\mathcal{P}_{T^\perp}(H_S)\|_F \\ &\leq \frac{1}{2}\|\mathcal{P}_\Omega(H_S)\|_F + \frac{1}{2}\|\mathcal{P}_{\Omega^\perp}(H_S)\|_F + \|\mathcal{P}_{T^\perp}(H_S)\|_F.\end{aligned}$$

Therefore,

$$\|\mathcal{P}_\Omega(H_S)\|_F \leq \|\mathcal{P}_{\Omega^\perp}(H_S)\|_F + 2\|\mathcal{P}_{T^\perp}(H_S)\|_F \leq \|\mathcal{P}_{\Omega^\perp}(H_S)\|_1 + 2\|\mathcal{P}_{T^\perp}(H_L)\|_*.$$

[4] That is, $Z_S = \lambda(\mathrm{sgn}(S_0) + F)$ with $\mathcal{P}_\Omega F = 0$ and $\|F\|_\infty \leq 1$; and $Z_L = UV^* + W'$ with $P_T W' = 0$ and $\|W'\| \leq 1$.

Combining the inequalities above, we have

$$\begin{aligned}\|X_0+H\|_\diamond &\geq \|X_0\|_\diamond + (1-\lambda/2-\|\mathcal{P}_{T^\perp}(\Lambda)\|)\|\mathcal{P}_{T^\perp}(H_L)\|_* \\ &\quad +(\lambda-\lambda/4-\|\mathcal{P}_{\Omega^\perp}(\Lambda)\|_\infty)\|\mathcal{P}_{\Omega^\perp}(H_S)\|_1 \\ &\geq \|X_0\|_\diamond + (3/4-\|\mathcal{P}_{T^\perp}(\Lambda)\|)\|\mathcal{P}_{T^\perp}(H_L)\|_* \\ &\quad +(3\lambda/4-\|\mathcal{P}_{\Omega^\perp}(\Lambda)\|_\infty)\|\mathcal{P}_{\Omega^\perp}(H_S)\|_1.\end{aligned}$$

□

Lemma 6. *Suppose that* $\|\mathcal{P}_T\mathcal{P}_\Omega\| \leq 1/2$. *Then for any pair* $X=(L,S)$, $\|\mathcal{P}_\Gamma(\mathcal{P}_T\times\mathcal{P}_\Omega)(X)\|_F^2 \geq \frac{1}{4}\|(\mathcal{P}_T\times\mathcal{P}_\Omega)(X)\|_F^2$.

Proof. For any matrix pair $X'=(L',S')$, $\mathcal{P}_\Gamma(X') = \left(\frac{L'+S'}{2}, \frac{L'+S'}{2}\right)$ and so $\|\mathcal{P}_\Gamma(X')\|_F^2 = \frac{1}{2}\|L'+S'\|_F^2$. So,

$$\begin{aligned}\|\mathcal{P}_\Gamma(\mathcal{P}_T\times\mathcal{P}_\Omega)(X)\|_F^2 &= \frac{1}{2}\|\mathcal{P}_T(L)+\mathcal{P}_\Omega(S)\|_F^2 \\ &= \frac{1}{2}\left(\|\mathcal{P}_T(L)\|_F^2+\|\mathcal{P}_\Omega(S)\|_F^2+2\langle\mathcal{P}_T(L),\mathcal{P}_\Omega(S)\rangle\right).\end{aligned}$$

Now,

$$\langle\mathcal{P}_T(L),\mathcal{P}_\Omega(S)\rangle = \langle\mathcal{P}_T(L),(\mathcal{P}_T\mathcal{P}_\Omega)\mathcal{P}_\Omega(S)\rangle \geq -\|\mathcal{P}_T\mathcal{P}_\Omega\|\|\mathcal{P}_T(L)\|_F\|\mathcal{P}_\Omega(S)\|_F.$$

Since $\|\mathcal{P}_T\mathcal{P}_\Omega\| \leq 1/2$,

$$\begin{aligned}\|\mathcal{P}_\Gamma(\mathcal{P}_T\times\mathcal{P}_\Omega)(X)\|_F^2 &\geq \frac{1}{2}\left(\|\mathcal{P}_T(L)\|_F^2+\|\mathcal{P}_\Omega(S)\|_F^2-\|P_T(L)\|_F\|P_\Omega(S)\|_F\right) \\ &\geq \frac{1}{4}\left(\|\mathcal{P}_T(L)\|_F^2+\|\mathcal{P}_\Omega(S)\|_F^2\right) = \frac{1}{4}\|(\mathcal{P}_T\times\mathcal{P}_\Omega)(X)\|_F^2,\end{aligned}$$

where we have used that for any a,b, $a^2+b^2-ab \geq (a^2+b^2)/2$. □

4.2.5 Proof of Proposition 4

Our proof uses two crucial properties of $\hat{X}$. First, since X_0 is also a feasible solution to (4.5), we have $\|\hat{X}\|_\diamond \leq \|X_0\|_\diamond$. Second, we use triangle inequality to get

$$\|\hat{L} + \hat{S} - L_0 - S_0\|_F \leq \|\hat{L} + \hat{S} - M\|_F + \|L_0 + S_0 - M\|_F \leq 2\delta.$$

Furthermore, set $\hat{X} = X_0 + H$ where $H = (H_L, H_S)$ and write $H^\Gamma = \mathcal{P}_\Gamma(H)$, $H^{\Gamma^\perp} = \mathcal{P}_{\Gamma^\perp}(H)$ for short. We want to bound $\|H\|_F^2$, which can be expanded as

$$\begin{aligned}\|H\|_F^2 &= \|H^\Gamma\|_F^2 + \|H^{\Gamma^\perp}\|_F^2 \\ &= \|H^\Gamma\|_F^2 + \|(\mathcal{P}_T \times \mathcal{P}_\Omega)(H^{\Gamma^\perp})\|_F^2 + \|(\mathcal{P}_{T^\perp} \times \mathcal{P}_{\Omega^\perp})(H^{\Gamma^\perp})\|_F^2. \end{aligned} \tag{4.9}$$

Since (4.9) gives us $\|H^\Gamma\|_F = \big(\|(H_L + H_S)/2\|_F^2 + \|(H_L + H_S)/2\|_F^2\big)^{1/2} \leq \sqrt{2}/2 \times 2\delta = \sqrt{2}\delta$, it suffices to bound the second and third terms on the right-hand-side of (4.9).

a. Bound the third term of (4.9)**.** Let W be a dual certificate satisfying (4.7). Then, $\Lambda = UV^* + W$ obeys $\|\mathcal{P}_{T^\perp}(\Lambda)\| \leq 1/2$ and $\|\mathcal{P}_{\Omega^\perp}(\Lambda)\|_\infty \leq \lambda/2$. We have

$$\|X_0 + H\|_\diamond \geq \|X_0 + H^{\Gamma^\perp}\|_\diamond - \|H^\Gamma\|_\diamond \tag{4.10}$$

and

$$\begin{aligned}\|X_0 + H^{\Gamma^\perp}\|_\diamond &\geq \|X_0\|_\diamond + (3/4 - \|\mathcal{P}_{T^\perp}(\Lambda)\|)\|\mathcal{P}_{T^\perp}(H_L^{\Gamma^\perp})\|_* \\ &\quad + \big(3\lambda/4 - \|\mathcal{P}_{\Omega^\perp}(\Lambda)\|_\infty\big)\|\mathcal{P}_{\Omega^\perp}(H_S^{\Gamma^\perp})\|_1 \\ &\geq \|X_0\|_\diamond + \frac{1}{4}\Big(\|\mathcal{P}_{T^\perp}(H_L^{\Gamma^\perp})\|_* + \lambda\|\mathcal{P}_{\Omega^\perp}(H_S^{\Gamma^\perp})\|_1\Big),\end{aligned}$$

which implies that

$$\|\mathcal{P}_{T^\perp}(H_L^{\Gamma^\perp})\|_* + \lambda\|\mathcal{P}_{\Omega^\perp}(H_S^{\Gamma^\perp})\|_1 \leq 4\|H^\Gamma\|_\diamond. \tag{4.11}$$

For any matrix $Y \in \mathbb{R}^{n\times n}$, we have the following inequalities:

$$\|Y\|_F \le \|Y\|_* \le \sqrt{n}\|Y\|_F, \frac{1}{\sqrt{n}}\|Y\|_F \le \lambda\|Y\|_1 \le \sqrt{n}\|Y\|_F,$$

where we assume $\lambda = \frac{1}{\sqrt{n}}$. Therefore

$$\begin{aligned}
\|(\mathcal{P}_{T^\perp} \times \mathcal{P}_{\Omega^\perp})(H^{\Gamma^\perp})\|_F &\le \|\mathcal{P}_{T^\perp}(H_L^{\Gamma^\perp})\|_F + \|\mathcal{P}_{\Omega^\perp}(H_S^{\Gamma^\perp})\|_F \\
&\le \|\mathcal{P}_{T^\perp}(H_L^{\Gamma^\perp})\|_* + \lambda\sqrt{n}\|\mathcal{P}_{\Omega^\perp}(H_S^{\Gamma^\perp})\|_1 \\
&\le 4\sqrt{n}\|H^\Gamma\|_\diamondsuit = 4\sqrt{n}(\|H_L^\Gamma\|_* + \lambda\|H_S^\Gamma\|_1) \\
&\le 4n(\|H_L^\Gamma\|_F + \|H_S^\Gamma\|_F) = 4\sqrt{2}n\|H^\Gamma\|_F \le 8n\delta, \qquad (4.12)
\end{aligned}$$

where the last equation uses the fact that $H_L^\Gamma = H_S^\Gamma$.

b. Bound the second term of (4.9)**.** By Lemma 6,

$$\|\mathcal{P}_\Gamma(\mathcal{P}_T \times \mathcal{P}_\Omega)(H^{\Gamma^\perp})\|_F^2 \ge \frac{1}{4}\|(\mathcal{P}_T \times \mathcal{P}_\Omega)(H^{\Gamma^\perp})\|_F^2.$$

But since $\mathcal{P}_\Gamma(H^{\Gamma^\perp}) = 0 = \mathcal{P}_\Gamma(\mathcal{P}_T \times \mathcal{P}_\Omega)(H^{\Gamma^\perp}) + \mathcal{P}_\Gamma(\mathcal{P}_{T^\perp} \times \mathcal{P}_{\Omega^\perp})(H^{\Gamma^\perp})$, we have

$$\begin{aligned}
\|\mathcal{P}_\Gamma(\mathcal{P}_T \times \mathcal{P}_\Omega)(H^{\Gamma^\perp})\|_F &= \|\mathcal{P}_\Gamma(\mathcal{P}_{T^\perp} \times \mathcal{P}_{\Omega^\perp})(H^{\Gamma^\perp})\|_F \\
&\le \|(\mathcal{P}_{T^\perp} \times \mathcal{P}_{\Omega^\perp})(H^{\Gamma^\perp})\|_F.
\end{aligned}$$

Combining the previous two inequalities, we have

$$\|(\mathcal{P}_T \times \mathcal{P}_\Omega)(H^{\Gamma^\perp})\|_F^2 \le 4\|(\mathcal{P}_{T^\perp} \times \mathcal{P}_{\Omega^\perp})(H^{\Gamma^\perp})\|_F^2,$$

which, together with (4.12), gives us the desired result,

$$\|H^{\Gamma^\perp}\|_F^2 \le 5\|(\mathcal{P}_{T^\perp} \times \mathcal{P}_{\Omega^\perp})(H^{\Gamma^\perp})\|_F^2 \le 64 \times 5 \times n^2\delta^2. \qquad (4.13)$$

4.2.6 Simulations

In this section, we run a series of numerical experiments on square matrices with noisy entries. For each setting of parameters, we report the average errors over 20 trials. Each entry of the noise term Z_0 is i.i.d. $N(0, \sigma^2)$. A rank-r matrix L_0 is generated as $L_0 = UV^*$ where both U and V are $n \times r$ matrices with i.i.d. $N(0, \sigma_n^2)$ entries, with $\sigma_n^2 \doteq 10\frac{\sigma}{\sqrt{n}}$. Here, the value of σ_n is rather arbitrary and set such that the singular values of L_0 are much larger than the singular values of Z_0. The entries of S_0 are independently distributed, each taking on value 0 with probability $1 - \rho_s$, and uniformly distributed in $[-5, 5]$ with probability ρ_s.

In order to stably recover $\hat{X} = (\hat{L}, \hat{S})$, instead of directly solving (4.5), we solve the following dual problem, to which a fast proximal gradient algorithm proposed in [65], *Accelerated Proximal Gradient* (APG), can be applied.

$$\min_{L,S} \ \|L\|_* + \lambda\|S\|_1 + \frac{1}{2\mu}\|M - L - S\|_F^2. \tag{4.14}$$

It is well established that (4.14) is equivalent to (4.5) for some value $\mu(\delta)$. Our choice of μ here follows similar arguments as in [20]. First, note that if we fix $S = 0$ in (4.14), the solution $\hat{L}$ of (4.14) is equal to the singular value thresholding version of M with threshold μ. Similarly, if we fix $L = 0$ in (4.14), the solution $\hat{S}$ is equal to the entry-wise shrinkage version of M with threshold $\mu\lambda$. Thus, we choose μ to be the smallest value such that the minimizer of (4.14) is likely to be $\hat{L} = \hat{S} = 0$ if we set $L_0 = S_0 = 0$ and $M = Z_0$. In this way, μ is large enough to threshold away the noise, but not too large to over-shrink the original matrices. Now, it is well known that for $Z_0 \in \mathbb{R}^{n\times n}$, $n^{-1/2}\|Z_0\| \to \sqrt{2}\sigma$ almost surely as $n \to \infty$. Thus, we choose $\mu = \sqrt{2n}\sigma$. This also fits the sparse component well since $\mu\lambda = \sqrt{2}\sigma$. We shall see that this choice of μ works well in practice.

Comparison with An Oracle. To further understand our algorithm, we would like to compare its performance to the best possible accuracy one can achieve, for instance, by the

minimal mean-square-error (MMSE) estimator over all low-rank and sparse matrix pairs. However, because obtaining the MMSE estimation is not computationally tractable, we instead resort to an oracle which gives us information about the support Ω of S_0 and the row and column spaces T of L_0. Our oracle estimates L and S as the solution L_{oracle} and S_{oracle} to the following least squares problem:

$$\min_{L,S} \|M - L - S\|_F \quad \text{subject to} \quad L \in T, S \in \Omega. \tag{4.15}$$

Since we know the locations of the corrupted entries, we can solve for L_{oracle} and S_{oracle} separately. That is, we first find the matrix in T which best fits the uncorrupted data in a least squares sense. Under the hypotheses of Theorem 4, the operator $\mathcal{P}_T\mathcal{P}_{\Omega^\perp}\mathcal{P}_T$ is invertible[5] when restricted to T and the least squares solution is given by

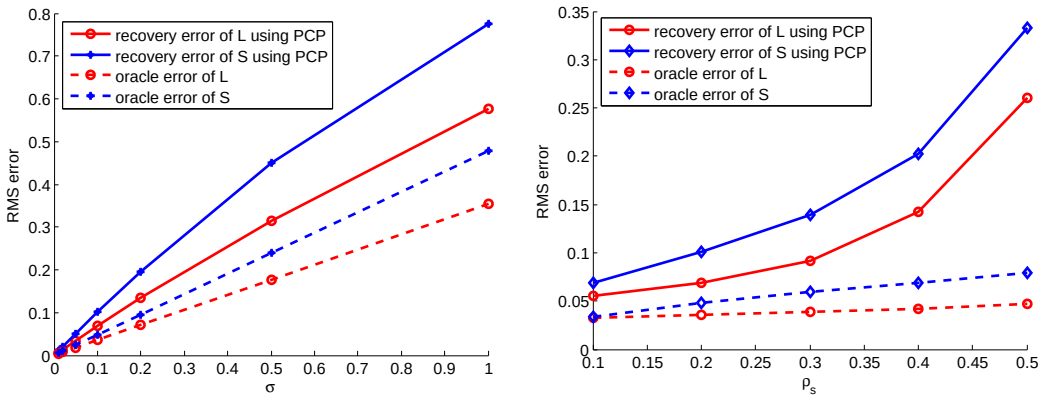
$$L_{oracle} \;=\; (\mathcal{P}_T\mathcal{P}_{\Omega^\perp}\mathcal{P}_T)^{-1}\mathcal{P}_T\mathcal{P}_{\Omega^\perp}(M),$$

and the sparse component is given by

$$S_{oracle} = \mathcal{P}_\Omega(M - L_{oracle}).$$

Experiment Results and Analysis. We first evaluate the performance of (4.14) with matrix L_0 whose rank $r = 10$ is fixed. We measure estimation errors using the root-mean-squared (RMS) error as $\|\hat{L} - L_0\|_F/n$, $\|\hat{S} - S_0\|_F/n$ for the low-rank component and the sparse component, respectively. Figure 4.1(a) shows the RMS error with varying noise level σ. In this experiment, the dimension $n = 200$ and the fraction of corrupted entries $\rho_s = 0.2$ are fixed. As predicted by our main result, the RMS error grows approximately linearly with the noise level. Moreover, the RMS error by solving (4.5) is just about twice the RMS error

[5]In fact, since $\|\mathcal{P}_T\mathcal{P}_\Omega\mathcal{P}_T\| = \|\mathcal{P}_\Omega\mathcal{P}_T\|^2 \le 1/4$, the smallest eigenvalue of $\mathcal{P}_T\mathcal{P}_{\Omega^\perp}\mathcal{P}_T$ is bounded below by $1 - 1/4 = 3/4$.



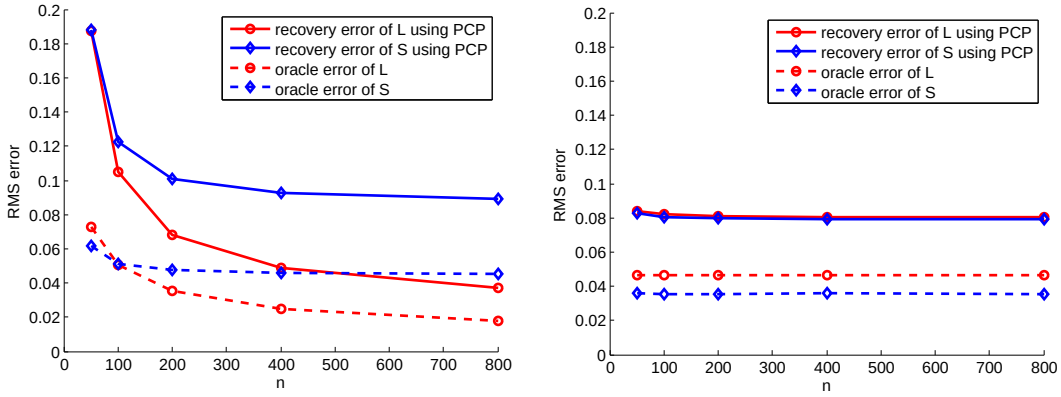
(a) (b)

Figure 4.1: (a) RMS errors as a function of σ with $r = 10, \rho_s = 0.2, n = 200$. (b) RMS errors as a function of ρ_s with $r = 10, \sigma = 0.1, n = 200$.

achieved by the oracle introduced in the previous section.

Now we fix $\sigma = 0.1$. Figures 4.1(b) and 4.2(a) show the results with varying ρ_s (when $n = 200$ is fixed) and n (when ρ_s=0.2 is fixed). Figure 4.1(b) illustrates that one can achieve higher breakdown point by knowing Ω and T. It is observed in [18] that when the rank r is fixed or grows sufficiently slowly as n increases, our method can recover more and more corrupted entries. Here in Fig. 4.2(a) we see a similar phenomenon. As n increases, the RMS error decreases given a fixed fraction of corrupted entries. That is, our approach can simultaneously tolerate a large fraction of corrupted entries and a high level of noise when the dimension n is sufficiently large.

To further test the stability of (4.14), we examine how the algorithm performs when the rank of L_0 grows in proportion to n and the fraction of errors in S_0 grows in proportion to n^2. More precisely, in Fig. 4.2(b) we fix $\sigma = 0.1$, and plot the RMS error as a function of n, with $\text{rank}(L_0) = 0.1 \times n$ and $\rho_s = 0.1$. The result clearly shows that our approach can recover a wide range of matrix pairs (L_0, S_0), in the presence of noise. Interestingly, these results also suggest that our analysis loses a factor of n with respect to the optimal bound.



(a) (b)

Figure 4.2: RMS errors as a function of n with (a) $\sigma = 0.1, \rho_s = 0.2, r = 10$ fixed, (b) $\sigma = 0.1, \rho_s = 0.1$ and $r = 0.1 \times n$ growing in proportion to n.

4.2.7 Discussion

In this chapter, we only present the result for square matrices for simplicity. However, the arguments and results can be easily modified to handle the general case. For instance, when the matrices are $n_1 \times n_2$, let $n_{(1)} = \max(n_1, n_2)$ and $n_{(2)} = \min(n_1, n_2)$. The conclusion of Theorem 1 can be stated as: PCP with $\lambda = 1/\sqrt{n_{(1)}}$ succeeds with probability at least $1 - cn_{(1)}^{-10}$, provided that $\text{rank}(L_0) \leq \rho_r n_{(2)} \mu^{-1} (\log n_{(1)})^{-2}$ and $m \leq \rho_s n_1 n_2$. Also, relation (4.6) in Theorem 2 becomes $\|\hat{L} - L_0\|_F^2 + \|\hat{S} - S_0\|_F^2 \leq C n_1 n_2 \delta^2$.

As suggested by the numerical results, one could hope to improve the stability result by removing the dependence on n. In this direction, we would like to point out that most of our analysis seems to be tight, except (4.12) where we invoke the generic relations between the nuclear norm, ℓ_1 norm and the Frobenius norm. Full examination of this problem may require additional model assumptions. It is also very likely that some results in the geometry of Banach spaces, namely the spherical sections theorem and concentration of measure, will play a key role in it.

Chapter 5

Holistic 3D Reconstruction from Low-Rank Textures

In Chapter 4, we described a new framework for low-rank matrix recovery via convex optimization and the major theoretical guarantees of its performance on both clean and noisy data. In this chapter, we apply the techniques described in Chapter 4 to explore symmetric or regular patterns in the images, and show how this leads to a holistic solution to 3D reconstruction of urban scenes without the use of any local features.

5.1 Introduction

Handling large-baseline images is a challenging problem in structure from motion. In the literature, in order to handle large-baseline images, which represent a large portion of images captured in popular applications [98, 42], sophisticated techniques have been proposed to extract and match image features beyond points and edges. Examples include affine-invariant features (SIFT) [70, 75, 78, 9], superpixels [79], and object part-based regions [121, 49], to name just a few. However, none of these features is truly invariant to camera viewpoint changes, and they are often sensitive to image errors.

In addition to improving local-feature detection, it has long been noticed that 3D reconstruction of urban structures can be more accurate and simple if one can detect in advance certain salient symmetric patterns (see [69] for a review on this topic) or global structures such as vanishing points [81]. However, symmetry and vanishing points are global or semi-global properties of the scene structures. They cannot be easily extracted from any individual image features. Instead, they must be inferred from the relations among a group of related

feature points or edges.

Despite numerous attempts [84, 69], it remains a challenging problem to reliably detect and extract large, symmetric patterns. The reason is twofold: First, it is difficult to properly detect all the features that represent a symmetric pattern (say the four corners and four edges of a window). Second, the task becomes more daunting in the presence of outliers and partial occlusion in the extracted feature set, which obscure the dominant global structures. This is the main reason robust statistical techniques such as Hough transform or RANSAC have been widely used for such purposes. Finally, even when local features are reliably extracted, it is not trivial to verify which ones satisfy what symmetric and/or vanishing point relations under camera perspective projection [55]. To address these problems, there has been an increasing amount of work trying to infer *approximate* 3D geometry of image patches of urban scenes using supervised machine learning methods [60, 52, 94]. In contrast, in this chapter, we investigate a novel approach to infer accurate 3D geometry from multiple large-baseline uncalibrated images of an urban scene in a mainly unsupervised fashion.

To avoid the aforementioned difficulties while inferring 3D geometry, we exploit a new class of "building blocks" for modeling urban objects. These new tools complement local features such as points, edges, SIFT features, and generic local patches. The new building blocks shall have the following good properties:

1. *Holistic*: They need to encode accurate, global geometric information such as structural symmetry, vanishing points, and camera positions.

2. *Invariant*: Their representation should be invariant to camera viewpoint and perspective distortion, so that they can be matched reliably across multiple images.

3. *Robust*: The extraction of such new features should be numerically stable and robust (say, to partial occlusion or random image noise and error).

Motivated by a new type of image feature called transform invariant low-rank texture (TILT) [118], in this chapter, we study how such low-rank textures can be used as new

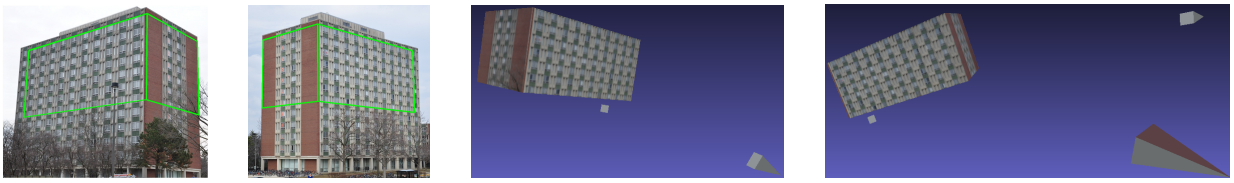


Figure 5.1: **Left Pair:** Example of matched facades of a building. **Right Pair:** Automatically reconstructed 3D model from *only four uncalibrated* images around the building by our method. Each image covers a pair of facades. The pyramids show the estimated location of cameras.

building blocks for modeling urban scenes. The proposed new approach suggests that we can obtain accurate 3D models for urban objects such as buildings and houses *without relying on extraction of any traditional local features such as points and edges*. The new approach relies directly and exclusively on semi-global or global image patches and regions built from TILT features. For this very reason, the approach is called "holistic." We show how to obtain accurate information about camera calibration, orientation and position from each image, correspondence between two images, and ultimately a consistent 3D structure from multiple images, as the example shown in Figure 5.1.

5.2 Geometry from One Facade of a Building

For completeness, we first give a brief review of work on low-rank textures [118] and then show how to use them for 3D modeling. It has been observed by the authors of [118] that the image of repetitive or symmetric patterns, when viewed as a matrix, is *low-rank*. For example, if I_0 is a rectified frontal view of a planar patch on the facade of a typical office building (see Figure 5.2 right), then as a matrix, I_0 has a rank much lower than its dimension. The authors call such an image patch *a low-rank texture*. In some other (perspective) view of the building (see Figure 5.2 left) the corresponding patch I deforms by a homography transform: $I = I_0 \circ \tau^{-1}$, where τ belongs to the homography group $GL(3)$ and deforms the image domain.

Figure 5.2: Geometry from a low-rank patch on a building facade. **Left:** The red box represents the selected candidate region I, and the green box corresponds to the recovered low-rank texture using TILT. **Right:** The rectified building facade $I_0 = I \circ \tau$ using the homography τ estimated from the low-rank texture.

One intriguing observation of the work [118] is that as long as the patch is large enough and contains sufficient texture, both the deformation τ and the view-invariant low-rank texture I_0 can be accurately recovered from the observed I, up to scaling in each of the image coordinates. The basic idea is to solve for a transformation τ of I so that $I_0 = I \circ \tau$ has the lowest possible rank. Furthermore, the image patch I is often corrupted by noise and occlusion. As a result, a more realistic model between the low-rank texture I_0 and its image I has been proposed by [118] as:

$$I \circ \tau = I_0 + E, \tag{5.1}$$

where E represents some sparse error that corrupts the image, say due to partial occlusion. As shown in the work [118] and Robust PCA literature [18], the transformation τ and the low-rank texture I_0 can be recovered by solving the following optimization problem:

$$\min_{A,E,\tau} \|A\|_* + \lambda \|E\|_1 \quad \text{subject to} \quad I \circ \tau = A + E, \tag{5.2}$$

where $\|\cdot\|_*$ and $\|\cdot\|_1$ represent the nuclear norm and ℓ_1-norm of a matrix, respectively [1].

The recovered low-rank texture A only differs from the original low-rank texture I_0 by a

[1]The nuclear norm of A is defined as the sum of its singular values: $\|A\|_* = \sum_i \sigma_i$. The ℓ_1-norm of E is defined as $\|E\|_1 = \sum_{i,j} |e_{ij}|$.

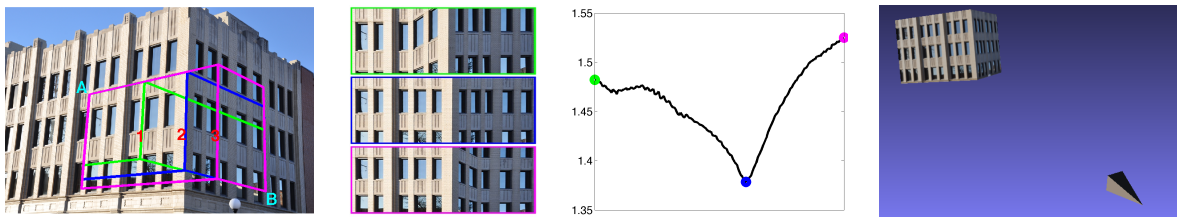
scaling in the x and y coordinates. The recovered τ encodes the homography from the default image plane $z = 0$ to the low-rank planar patch in 3D: $\tau^{-1} \doteq [\boldsymbol{t}_1, \boldsymbol{t}_2, \boldsymbol{t}_3] = K[\boldsymbol{r}_1, \boldsymbol{r}_2, T]$, where $R = [\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3] \in \mathbb{R}^{3\times3}$ is the rotation, $T \in \mathbb{R}^3$ the translation, and $K \in \mathbb{R}^{3\times3}$ the intrinsic parameters of the camera. If the horizontal and vertical directions of the low-rank patch are parallel to two vanishing directions in 3D, then the first and second columns of τ^{-1} as a 3×3 matrix give the coordinates of the two vanishing points $\boldsymbol{v}_1 = \boldsymbol{t}_1, \boldsymbol{v}_2 = \boldsymbol{t}_2 \in \mathbb{R}^3$ in the image coordinates, respectively [72]. If the camera is calibrated, the two vanishing points should be orthogonal to each other. So from a low-rank texture region in an uncalibrated image, we obtain one linear constraint on the camera intrinsic parameters $K \in \mathbb{R}^{3\times3}$: $\boldsymbol{v}_1^T K^{-T} K^{-1} \boldsymbol{v}_2 = 0$. If the image(s) consist of a sufficient number (≥ 5) of low-rank patches with independent orientations in 3D, one can fully recover the camera intrinsic parameters K without any special calibration apparatus.

5.3 Geometry from Intersecting Facades

Although the TILT method allows us to extract geometry from each individual low-rank patch, an urban scene typically consists of numerous low-rank regions. A representative image of a building may contain two or more of its facades, as shown in Figure 5.3(a). The homographies recovered from individual patches on each of the facades may not be consistent in their scales.

Normally the low-rank textures on two intersecting facades of a building give three sets of parallel lines, two horizontal and one vertical. These three sets of parallel lines define three vanishing points in the image, denoted as $\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3 \in \mathbb{R}^3$, respectively. Notice that the pairs $(\boldsymbol{v}_1, \boldsymbol{v}_3)$ and $(\boldsymbol{v}_2, \boldsymbol{v}_3)$ can be obtained from the homography of an individual low-rank patch on each of the facades.

In order to determine the relative scale of the two facades in 3D, we need to find their intersection line l in the image. It belongs to the one-parameter family of lines that go



(a) (b) (c) (d)

Figure 5.3: Identifying the intersection line l of two facades. **(a)** Three different intersection hypotheses for the two adjacent 4-sided polygons. **(b)** The unfolded joint textures: each corresponds to one of the hypotheses shown in (b), as indicated by the color of its boundary. **(c)** The value of (5.3) as a function of the location of the intersection line. It is minimized precisely at the correct location (the blue line).**(d)** Accurate 3D "pop-up" from this single image. Camera position is recovered, shown as a small pyramid.

through the vanishing point $\boldsymbol{v}_3$ in the image. As it turns out, we can use the *joint low-rank structure* of both facades to determine the precise location of this line regardless whether there is a visible edge along this line or not.

To see this, let us fix one point in each facade, say, the upper-left corner of a low-rank patch on the first facade and the lower-right corner of a patch on the second facade, labeled as points $\boldsymbol{A}$ and $\boldsymbol{B}$ respectively, as shown in Figure 5.3(a). As one can see, since the vanishing points $\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3$ are known, any intersection line l between $\boldsymbol{A}$ and $\boldsymbol{B}$ will uniquely determine a special structure with two adjacent 4-sided polygons in the image, each corresponding to a facade of the building. That is, the homographies τ_1 and τ_2 of the two facades are parametrized by the same one-parameter family lines l passing through $\boldsymbol{v}_3$. Figure 5.3(a) shows examples of the special structure with three different intersection lines (labeled as 1, 2 and 3).

Given the corresponding homography $\tau_i(l)$, we may rectify each polygon and then concatenate them into a joint rectangular texture, as shown in the Figure 5.3(b). Then the joint texture, as a matrix, should also have the lowest rank when the intersection line is the correct one (Figure 5.3(c)).

Mathematically, let I_1 and I_2 be the two low-rank texture windows subject to transformations τ_1 and τ_2, which depend only on l. We find the true position of the intersection line

l^* by solving the following optimization problem:

$$\begin{aligned} l^* &= \arg\min_l \|[A_1\ A_2]\|_* + \lambda\|[E_1\ E_2]\|_1 \\ &\text{s.t.} \quad I_i \circ \tau_i(l) = A_i + E_i, \quad i = 1, 2. \end{aligned} \tag{5.3}$$

This problem can be solved very effectively using a line search technique along the unknown parameter l. Figure 5.3(c) shows a typical plot of values of the objective function. Once the intersection line l^* is determined, the relative scale of the two facades and camera geometry are uniquely determined. Figure 5.4 shows more representative results. As one can see, the proposed scheme accurately identifies the correct intersection lines even when local edge features around the intersection, on which most traditional methods rely, are almost *invisible* in the image (e.g., in Figure 5.4(b)) or even suggest an incorrect line (e.g., in Figure 5.4(c))!

If the camera is calibrated, from the assumption we know $\boldsymbol{v}_3$ should be orthogonal to both $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ as $\boldsymbol{v}_3 \sim \boldsymbol{v}_1 \times \boldsymbol{v}_2$. Very often, the two facades of the building are also orthogonal to each other, i.e., $\boldsymbol{v}_1 \perp \boldsymbol{v}_2$.[2] If the camera is not calibrated, the three vanishing points impose three independent constraints on the camera intrinsic parameters:

$$\boldsymbol{v}_1^T K^{-T} K^{-1} \boldsymbol{v}_2 = 0, \boldsymbol{v}_1^T K^{-T} K^{-1} \boldsymbol{v}_3 = 0, \boldsymbol{v}_2^T K^{-T} K^{-1} \boldsymbol{v}_3 = 0. \tag{5.4}$$

This allows us to fully calibrate the camera from just a pair of intersecting facades, if only the focus length f and principal point (o_x, o_y) are not known in K. An example of such reconstruction from single image is shown in Figure 5.3 (d).

5.4 Segmenting Building Facades

Patches of low-rank textures allow us to extract from a single image accurate information about the camera location, calibration, and 2D textures and 3D structures. But in order

[2]This may not always be the case. For instance, the facades in Figure 5.9 (a) and (b) are not orthogonal.

(a) (b) (c) (d)

Figure 5.4: Additional representative results of identifying the intersection line of two adjacent facades. Red windows are the initialization.

to obtain a complete 3D model from multiple images around a large building, we need to establish correct, precise point-wise correspondence between different views.

Repetitive features and patterns in an urban scene make finding the correct correspondence between images much more challenging than that for a generic non-urban scene. The reason is obvious: Matching local features or even local patches are inherently ambiguous – there are many other points and patches in the other image(s) that have exactly the same local appearance. Most SFM methods then rely on having images taken with relatively small baselines, either from a video sequence or from a very dense set of photos.

When the baseline between images is large or images are sparse, any effort to eliminate such ambiguity has to rely on certain global spatial relationships among multiple points, lines, or patches. The approach we propose here relies on a very simple observation: *the larger the patch or region we match, the less the ambiguity* [111, 110]. To the extreme, if we can detect the entire facades, then the matching would have minimal ambiguity. Hence, a necessary step to establish globally consistent correspondence between views is to segment out each building facade.

As different facades of the same building often have the same local color and textural appearance (see Figure 5.4), global geometry and texture become the only cues to tell them apart. Our approach relies on another simple observation: *if two adjacent patches, say* I_1, I_2*, belong to the same facade, then after we merge them into a larger patch* $I = [I_1, I_2]$*, the joint texture should remain low-rank* (after rectification by a homography found by TILT:

$I \circ \tau = A + E$). Such a patch I can be represented very compactly by the triplet (A, E, τ): the homography τ, the low-rank component A, and the sparse component E. Thus, by comparing the compactness of the representation before and after the merging, we can tell whether the two patches belong to the same facade or not.

In the rest of the section, we first derive a purely objective measure for the compactness of a patch I based on its coding length,[3] and then we show how to use this measure to effectively cluster patches to form facades.

5.4.1 Compact Coding for Low-rank Textures

A naive way to encode the patch I would be entropy-coding of the quantized sequence of pixel values in I, as conventional image compression schemes do. However, when rank(A) is small and E sparse, encoding I in terms of the triplet (A, E, τ) is far more efficient as both sparse and low-rank matrices allow efficient coding. In order to get a finite coding length, the components of the triplet must be quantized. Denote the number of bits required to represent a quantized real number by f.[4] For controlling overall reconstruction quality of the patch, we define a distortion parameter ϵ. No matter how we encode the patch, the decoded triplet $(\hat{A}, \hat{E}, \hat{\tau})$ must satisfy a distortion tolerance:

$$\|(\hat{A} + \hat{E}) \circ \hat{\tau}^{-1} - I\|_F^2 \leq \epsilon^2 size(I), \tag{5.5}$$

where $size(I)$ is the number of pixels of I, say $m \times n$.

[3] There is a theoretical connection between rank and the coding length of a matrix [91]. However, rank is very sensitive to noise and outliers. We have conducted experiments using the aggregated rank, and the segmentation results are unstable. The proposed coding length is essentially a robust measure of rank based on Robust PCA of the image region.

[4] We have empirically observed that for any real number, 16 bits are more than sufficient to ensure a good precision. For example, the homography τ is a 3×3 matrix. Thus, it is sufficient for us to assign $9f$ bits to it, i.e. $L(\hat{\tau}) = 9f$, where $\hat{\tau}$ is the quantized τ.

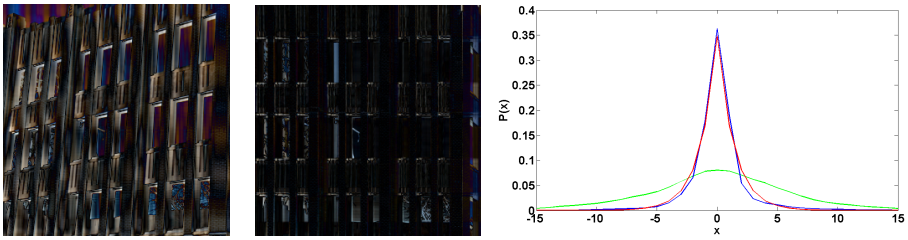


Figure 5.5: **Left:** The residual matrix $\hat{E}$ of the original (left) and transformed (right) images in Figure 5.6 approximated by top three singular values/vectors. **Right**: Empirical probability distributions of the errors for the left (green) and right (blue) residual maps. The empirical distribution (blue) of the right residual map can be fit closely by a Laplace distribution (red).

Encoding the Sparse Matrix E. The sparsity in $\hat{E}$ implies that it has a very low-entropy – many entries are zero. It has long been observed empirically in signal processing that most sparse signals obey a *Laplace distribution* [16]: $p(x) = \frac{1}{2\lambda} \exp\big(-\frac{|x-\mu|}{\lambda}\big)$, where we typically assume $\mu = 0$ in our setting. Since we here are working with a set of discrete samples: $\mathcal{X} = \{x_1, \ldots, x_N\}$[5], we can work with a discrete Laplace distribution $p_k = \frac{1}{Z\Lambda} \exp\big(-\frac{|x_k|}{\Lambda}\big)$, over some support interval $[-B, B]$. Here Z is the normalization constant and x_k is a sampling point. We simply choose $B = \max_i |x_i|$ over the sample set $\mathcal{X}$. The maximum likelihood estimate of Λ based on $\mathcal{X}$ is given by the following expression: $\Lambda = \frac{1}{N} \sum_{i=1}^{N} |x_i|$.

Figure (5.5) shows a typical example of empirical distribution of $\hat{E}$ (blue), from one of the building images, against the estimated distribution $\{p_k\}$ (red). The distribution $\{p_k\}$ has two parameters, namely (B, Λ). Thus, by merely transmitting B and Λ, which takes $2f$ bits, the receiver can construct $\{p_k\}$ and use it to infer the optimal codebook for $\mathcal{X}$. With such a (Laplace) encoder, the expected coding length for $\hat{E}$ would be:

$$L(\hat{E}) = 2f + mn(\sum_{k} -p_k \log_2 p_k). \tag{5.6}$$

Encoding the Low-rank Matrix A. Naive entry-wise encoding of the $m \times n$ quantized low-rank matrix $\hat{A}$ would take mnf bits. However, since A is low-rank, the singular value

[5]Each $x_i \in \mathcal{X}$ is an element of the matrix $\hat{E}$, thus $|\mathcal{X}| = N = mn$.

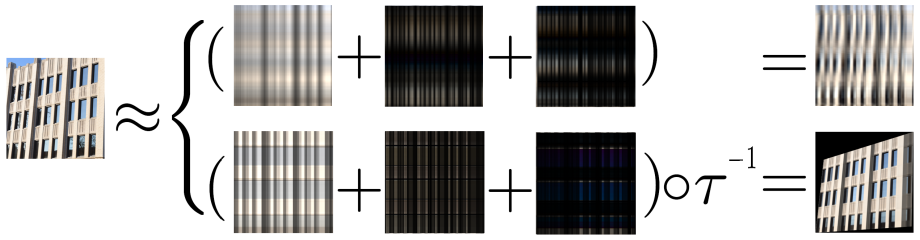


Figure 5.6: Approximation of a facade image with the top three singular components $\sum_{i=1}^{3} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T$. **Top:** SVD of the original image. **Bottom:** SVD of the rectified image by TILT.

decomposition $A = U\Sigma V^T$ leads to a more effective encoding. Let $r = \text{rank}(A)$. Then, we only need to encode $(m + n + 1)rf$ bits associated with (quantized) non-zero singular values and their corresponding singular vectors: $\hat{A} = \sum_{i=1}^{r} \hat{\sigma}_i \hat{\boldsymbol{u}}_i \hat{\boldsymbol{v}}_i^T$, where the non-quantized variables are $\boldsymbol{u}_i \in \mathbb{R}^m$, $\boldsymbol{v}_i \in \mathbb{R}^n$ and $\sigma_i \in \mathbb{R}_+$. Obviously, for $r \ll \min\{m, n\}$, this encoding uses much fewer bits than the naive encoding $(m + n + 1)rf \ll mnf$.

For noisy real images, A may not be a perfectly low-rank matrix. So we only need to encode its leading rank-q approximation: $A_q = \sum_{i=1}^{q} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T$ subject to the allowed distortion ϵ. The coding length of $\hat{A}_q$ is thus given by:

$$L(\hat{A}_q) = (m + n + 1)qf. \tag{5.7}$$

We can further compress the vectors $\{\hat{\boldsymbol{u}}_i, \hat{\boldsymbol{v}}_i\}$ based on additional structures in them. As each $\hat{\boldsymbol{u}}_i$ or $\hat{\boldsymbol{v}}_i$ is often a smooth signal except at the image edges (see Figure 5.6), we can encode each vector by a *difference* code[6] plus the head element. This way, the difference of each vector is a sparse sequence and can be encoded again by the Laplace code.

[6]The code subtracts from each element in the sequence the value of the previous element.

(a) (b) (c) (d) (e) (f)

Figure 5.7: **(a)** Initial grid. **(b)** Initial TILT. **(c)** Final segmented regions. **(d)** Recovered intersection line. **(e)-(f)** The homography estimated from cyan and magenta regions applied to the entire I to get the transformed images I' (corresponding regions are rectified).

5.4.2 Compression-Based Facade Segmentation

To summarize, the coding length required to encode a supposedly low-rank patch I subject to the distortion tolerance ϵ is given by:

$$\min_{q} L(\hat{A}_q) + L(\hat{E}) + L(\hat{\tau}) \text{ s.t. } \|(\hat{A}_q + \hat{E}) \circ \hat{\tau}^{-1} - I\|_F^2 \leq mn\epsilon^2, \tag{5.8}$$

where $(\hat{A}, \hat{E}, \hat{\tau})$ is the decoded quantized version of (A, E, τ).

For an image that contains multiple facades, we segment the image I into a set of subregions $\mathcal{S} = \{I_k\}$ whose union covers all the valid TILT features in I. The goal is to choose $\mathcal{S}$ such that, when each I_k is encoded by the proposed scheme, the total coding length becomes minimal:

$$\begin{aligned} &\min_{\mathcal{S},\{q_k\}} \quad \textstyle\sum_k L(\hat{A}_{k,q_k}) + L(\hat{E}_k) + L(\hat{\tau}_k) \\ &\text{s.t.} \quad \|(\hat{A}_{k,q_k} + \hat{E}_k) \circ \hat{\tau}_k^{-1} - I_k\|_F^2 \leq mn\epsilon^2, \ \forall k. \end{aligned} \tag{5.9}$$

We solve this problem in a greedy and agglomerative fashion, similar to that in [91]. The algorithm starts from a simple grid on I, where each I_k is a tile of the grid. At each subsequent iteration, a pair of adjacent regions are chosen to be merged into a larger region, which leads to maximal reduction in the total coding length (5.9). The process stops when the number of bits can no longer be reduced given the distortion. Figure 5.7 shows some representative results.

Comparison with Symmetry Detection. Conceptually, one could also utilize the work of [84] to parse the building facades, which can effectively detect and segment regions tiled by a repetitive 2D pattern. We have tested their method on our data and found that the method is not suitable for our purposes due to several reasons: it often breaks one facade into multiple disconnected small lattices; the symmetry groups/lattices detected from different images (of the same facade) can be very different, and it cannot handle large perspective distortion. These problems make the results hard to use for subsequent matching.

5.5 Point-wise Matching of Building Facades

The segmentation provides a good estimate for the relative location of the facades and their rectified texture (see Figure 5.7 (e) and (f)). Using such rectified textural regions, solving large-baseline correspondence between two images I_1 and I_2 becomes better conditioned (say by a similarity match). However, each segmented region may not share the same location and scale in different images. Therefore, we need to refine their location and scale in order to obtain precise point-wise matching between images.

Denote A_1 as a low-rank texture from one facade in the first image I_1. If we assume the triplet (A_2, E_2, τ_2) in the second image I_2 best matches A_1 among all obtained segments in I_2, then the entire image I_2 can be rectified by the homography τ_2, and the sparse error E_2 be removed before matching. Thus, the problem is reduced to matching A_1 to a cleaned image: $I_2' = I_2 \circ \tau_2 - E_2$ (see Figure 5.7).

The goal now is to find a region R^* in I_2' which, after translation and scaling, best matches A_1 point-wise. We use normalized cross correlation (NCC) to measure the similarity between the two regions, which is ideal for our task as the regions are already distortion-free. Therefore, the best region is given by the following optimization:

$$R^* = \arg \max_{\phi=(x,y,u,v)} \frac{\operatorname{vec}(A_1)^T \operatorname{vec}(R \circ \phi)}{\| \operatorname{vec}(A_1)\|_2 \| \operatorname{vec}(R \circ \phi)\|_2}, \tag{5.10}$$

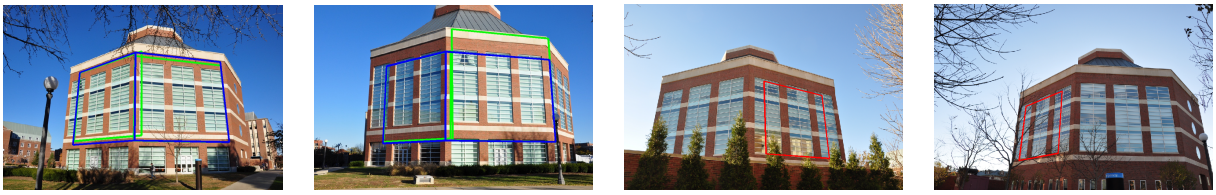
(a) (b) (c) (d) (e)

Figure 5.8: **(a)** Segmented and unwarped facade. **(b),(c)** Segmented and unwarped region of the same facade in a different image. In (c), the segmentation result is further refined to the orange box by matching. **(d)** Point-wise match between two regions of the facades. **(e)** Feature-point matching result of the two rectified regions by SIFT [70], with red lines indicate mismatches.

where ϕ is parameterized by the center location (x, y) of R and scales (u, v) in x and y directions, respectively.

We solve the optimization task iteratively. Initially, we start from a guess (x_0, y_0, u_0, v_0), which is a box among the candidate regions in I_2 (such as those in Figure 5.7) that has the highest NCC with A_1. We then maximize the objective function in a gradient ascent fashion. The iteration terminates when no more improvement can be made. Due to the greedy nature of this procedure, theoretically we can only guarantee a local optimal matching region $\hat{R}$. However, since we are working with very large segmented regions, we have observed in practice that the above procedure typically finds the globally optimal matching. Again, since there is no geometric distortion left in the rectified low-rank textures, the refinement converges to a very precise point-wise matching. If the two images each has multiple (segmented) facades, we run the above matching procedure on each candidate pair and choose the one that has the best matching score. As the number of segments is typically very small (2 or 3 per image in most cases), this process is very efficient.

Comparison with Feature Matching. An example of final matching results between two images is given in Figure 5.8. As a comparison, in Figure 5.8 (e), we illustrate the difficulty of applying the classical SIFT matching technique [70] to the same urban scenes with repetitive or symmetric patterns. Point-wise matching of low-rank regions outperforms SIFT in this scenario because the texture segmentation results enable us to perform accurate



(a) (b) (c) (d)

Figure 5.9: **(a) and (b)** Segmentation (green) and intersection detection (blue) on two images of an octagonal building. **(c) and (d)** A pair of matched regions from the same facade with different partial occlusion.

region-based matching rather than using local points or edges.

5.6 Full 3D Reconstruction of Buildings

In this section, we demonstrate how the techniques from the earlier sections can be assembled together for 3D reconstruction of a large octagonal building.[7] We use only eight *uncalibrated and widely separated* images for the full reconstruction of the building. Each of the images covers a pair of adjacent facades as shown in Figure 5.9. This building has a few interesting properties. First, the large number of facades and intersections will magnify the accumulation of (geometry or calibration) error if any. Second, occlusion by trees and reflections on the glass are two major problems that challenge conventional SFM methods, but can testify the *robustness* of our scheme against such errors.

We do not use any prior information about the geometric model of the building except that all the facades share the same vertical vanishing point. We use the vanishing point constraints to partially determine the calibration matrices of the eight images. Since two facades in each image impose two independent constraints on the calibration matrix, we use them to recover the focal length f and the x-coordinate o_x of the principal point, assuming the y-coordinate o_y is fixed at one half of the image height. Once the calibration matrix is obtained, we can compute the relative orientation and position of the camera with respect

[7] For 3D reconstruction of a typical rectangular building, see Figure 5.1.

to the scene.

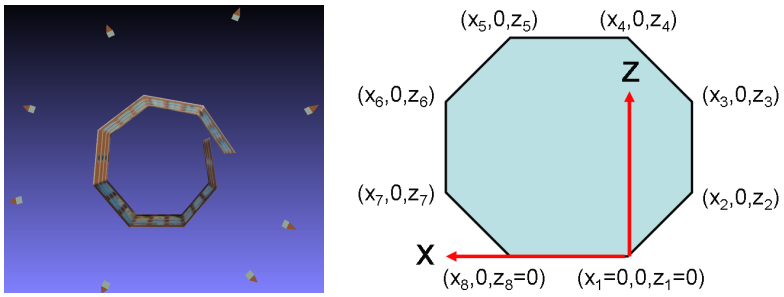
To segment the facades, we assume the rough location of the building within the images is provided.[8] A 5×5 grid of initial windows is then placed around this location. Some of the identified facades for the octagonal building are shown in Figure 5.9(a) and (b). We further arrange the sequence of images so that matching of common facades is only performed between consecutive images. See Figure 5.9(c) and (d) for an example of the matched facades.

Now we can obtain a full 3D reconstruction by assembling the building one view at a time using consecutively matched facades. However, errors in both camera parameters and the 3D model, when estimated from real images, are inevitable. For example, the camera calibration may not be precise enough because of simplifying assumptions on the parameters (i.e., f, o_x, o_y). Thus, if we assemble the views one by one, geometric error accumulates as the number of images increases. For example, the start and the end of the model do not meet each other in Figure 5.10(a).

Enforcing Global Consistency. For global consistency, we propose a global objective, which uses the current camera parameters and 3D model as the input, and tries to refine them simultaneously. Conceptually, this is similar to "bundle adjustment" in conventional SFM.

We randomly select two adjacent facades, say the pair in Figure 5.9(a), and choose the origin of the world frame to be a point at the intersection of the two facades. In addition, we let the x and y axes of the world frame to be parallel to the left facade in that image. Once the world frame is chosen, a building with n facades can be described using a set of n points $X = \{X_i\}_{i=1}^n$, where each $X_i = (x_i, 0, z_i)^T$ is (1) on the plane $y = 0$ and (2) at the intersection line of two adjacent facades. These points form a n-sided polygon on the $y = 0$ plane. For example, the 3D model of the octagonal building ($n = 8$) is shown in

[8] Either by the user or by a simple detection scheme.



(a) (b)

Figure 5.10: **(a)** A top view of the reconstructed structures of the octagonal building showing the accumulated geometry error when assembling the views one by one. **(b)** The parameterized 3D model of the building.

Figure 5.10(b).

For the cameras, we use the same set of parameters $\{K_i, R_i, T_i\}_{i=1}^n$ as before. Here we assume both the focal length f_i and the principal point (o_{x_i}, o_{y_i}) of each camera are unknown. Now we formulate the global optimization as follows. First, from each image I_i, we can extract two rectified facades $(A_i^j, E_i^j), 1 \leq j \leq 2$: $I_i \circ \tau_i^j(K_i, R_i, T_i, X) = A_i^j + E_i^j$. Second, we ask the i-th pair of matching facades to be the same, up to some sparse error $\boldsymbol{e}_i$:

$$I_i \circ \tau_i^2(K_i, R_i, T_i, X) = I_{i'} \circ \tau_{i'}^1(K_{i'}, R_{i'}, T_{i'}, X) + \boldsymbol{e}_i, \tag{5.11}$$

where $i' = \mod(i+1, n)$. Combining these two criteria, we propose to solve the following problem:

$$\begin{aligned} \min \sum_{i=1}^{n} \sum_{j=1}^{2} \left\{ \|A_i^j\|_* + \lambda \|E_i^j\|_1 \right\} + \sum_{i=1}^{n} \gamma \|\boldsymbol{e}_i\|_1, \\ \text{s.t.} \quad I_i \circ \tau_i^j(K_i, R_i, T_i, X) = A_i^j + E_i^j, \\ I_i \circ \tau_i^2(K_i, R_i, T_i, X) = I_{i'} \circ \tau_{i'}^1(K_{i'}, R_{i'}, T_{i'}, X) + \boldsymbol{e}_i, \end{aligned} \tag{5.12}$$

where λ and γ are the weights of the respective term. To deal with the nonlinear constraints in (5.12), we use an iterative scheme, which repeatedly solves the linearized version of (5.12) w.r.t. the current estimates of all unknown parameters $(K_i, R_i, T_i, X_i)_{i=1}^n$. To reduce the

Figure 5.11: Frontal (left & middle) and top (right) views of the recovered building. Each pyramid shows the estimated location of a camera.

effect of change in illumination and contrast, we normalize each $I_i \circ \tau_i^j$ to zero mean and unit Frobenius norm. With the initialization obtained from assembling the views one by one, the iterative scheme usually converges in 15 to 20 iterations.

Figure 5.11 shows the reconstructed full 3D model as well as the recovered camera poses. The readers should note the improvement in the top view of the 3D model, compared to Figure 5.10(a). We also calculated the average error in the eight angles between the building facades. It is 3.1 degree and 1.5 degree before and after global adjustment, respectively. As one can see, despite unknown calibration, partial occlusion, large baselines, our method is able to recover a very precise and complete 3D model of the building.

Comparison with other SFM Systems. It is difficult to make a fair comparison between the proposed approach and other SFM methods, since the large baselines and rich symmetry make other methods fail. In fact, we tested our sequences on almost all publicly available SFM packages such as Bundler [98], SFM-SIFT[9] (which combines Torr's SFM toolbox [106] with SIFT feature detector [70]), FIT3D [37], and Voodoo Camera Tracker.[10] All these packages report errors related to their inability of establishing meaningful correspondence across the views.

[9] `http://homepages.inf.ed.ac.uk/s0346435/projects/sfm/sfm_sift.html`
[10] `http://www.digilab.uni-hannover.de/docs/manual.html`

Chapter 6

Low-rank Panoramas for Street View Videos

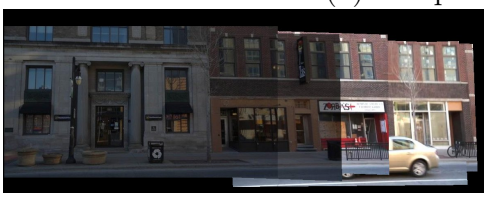
In Chapter 5, we saw how low-rank matrix recovery techniques could be effectively employed to exploit the internal regularities of each image, resulting in a set of simple yet powerful tools for 3D reconstruction of urban scenes. In this chapter, we will apply the low-rank matrix recovery techniques to harness the regularities and redundancy among a large number of frames of a video. Compared to a set of unsorted large-baseline images, frames of a video are more closely correlated, hence allowing us to recover the underlying low-dimensional structures in a more aggressive and efficient fashion. To this end, we propose a new method to generate street view panoramas from videos and demonstrate its robustness to illumination changes, occlusions and dynamic objects in the scene.

6.1 Introduction

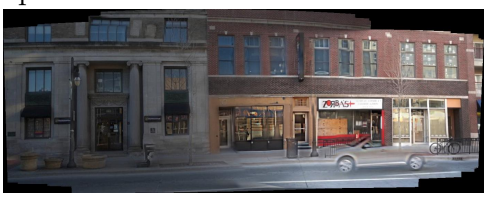
Recently, driven by industrial applications such as map building, virtual reality, and automatic navigation in urban environments, there has been tremendous interest in and effort toward building large-scale textured geometric models for urban areas from *street view videos*, taken by a moving camera mounted on cars (see Figure 6.1). One of the common and challenging tasks for effectively compressing and presenting such video data is to align and stitch a long video sequence to generate a single seamless panorama for the street view.

Image stitching (or mosaicing) is often tackled with a two-step approach. The first step is image alignment, which aims to map corresponding pixels from one image to another. In the literature, a variety of parametric deformation models are available for this purpose. For

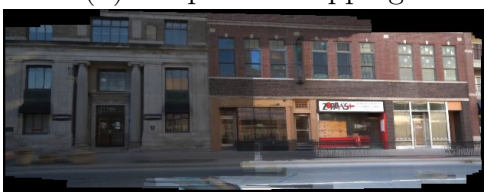
(a) Sampled Input Frames



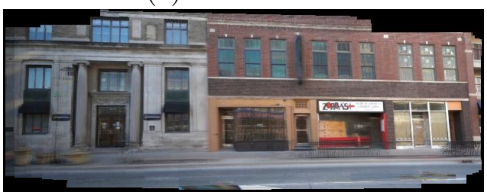
(b) Simple Overlapping



(c) AutoStitch



(d) Median Filter



(e) Our Result

Figure 6.1: Comparison of stitching results of a street view video. **(a)** The input sequence is challenging due to camera exposure changes and moving objects (e.g., cars) in the scene. **(b)** Simply overlapping video frames together creates visible seams and cutting-through effects in the result. **(c)** Current state-of-the-art algorithms such as AutoStitch often suffer from the undesired ghosting effect caused by the occluding objects. **(d)** While being robust to outliers, median filter cannot handle inconsistent exposures or illuminations. **(e)** Both moving objects and exposure changes are properly handled by our method, which generates a clean panorama.

street view videos, we typically want to build a panorama for the building facade along the street, which can be modeled approximately as a plane. Thus, in this chapter, we adopt the common *planar perspective motion model* for aligning video frames into the same coordinate system. This consists of estimating a homography (general linear transformation of a 2D plane) between every two consecutive frames. Figure 6.1(b) shows an example of overlapping the video frames in Figure 6.1(a) according to the estimated homographies.

As one can see in Figure 6.1(b), even with accurate alignment, the stitching result by simple overlaying looks very awkward. The reason is at least two-fold: First, the scene appearance or camera parameters (e.g., exposure) often change during the recording process, leading to visible seams in the stitching result. Second, the moving objects (e.g., cars) in the scene violate the (planar homography) model, resulting in the cutting-through or ghosting

effect in Figure 6.1(b). Therefore, the second step of most conventional approaches is to stitch the frames together using some blending algorithms to alleviate these problems, and generate *a single panorama image that tries to keep all the objects in the scene*. Unfortunately, this goal turns out to be impossible as not all objects can be fit with the same global parametric model. As a result, even for a panorama that is generated by one of the state-of-the-art algorithms AutoStitch [15] in Figure 6.1(c), there are obvious problems: One can easily notice the unbalanced illumination in the panorama, and the ghost effect due to its failure to handle occluding or moving objects effectively.

Hence, different from the conventional methods, in this chapter we propose to simply remove from the final panorama all the objects that do not fit the parametric motion model (i.e., outliers), which include occluding objects, reflection on glasses, or some large parallax in the scene (see the supplementary video). Note that if outliers are the only problem and the same point shows up in all images with constant intensity, a median filter would often do a good job removing the outliers from the panorama. However, as shown in Figure 6.1(d), it cannot handle inconsistent exposures or illuminations, and hence produces visible seams in a long sequence.

This observation motivates us to reconsider the following question: Is it really necessary to blend all the input images into a single mosaic image? In fact, each input image can be viewed as an incomplete (windowed) view of a scene under certain unknown exposure or illumination. The key observation here is that if we can complete each view then we have multiple complete views of the scene under different exposures and illuminations. Most importantly, it has been shown that these views all lie in *a low-dimensional subspace*, not only for Lambertian scenes [8], but for scenes under very general real-world lighting and viewing conditions as well [45], except for some outlying regions (reflections or occlusions). We call each element within this subspace a *view-dependent panorama* of the scene. Hence, instead of blending all the input images into a mosaic image, we argue that it is more appropriate to recover a view-dependent panorama for each of the input image (see Figure 6.2).

(a) (b) (c) (d)

Figure 6.2: View-dependent panoramas from a street video sequence. **(a)** Input frames. **(b)** Residual images. **(c)** Recovered view-dependent panoramas for the input frames. **(d)** Details of a small region in (c).

Mathematically, given a sequence of aligned input images, the problem is equivalent to recovering a low-rank matrix (one column per input image) with severely under-sampled (windowed) measurements which are partially corrupted. In this chapter, we leverage new convex optimization tools for recovering low-rank and sparse signals [92, 21, 18, 26, 113, 115] and develop an efficient and scalable algorithm for solving this problem effectively. Somewhat surprisingly, we show that, with such a benign low-rank assumption, it is possible to recover all missing parts of each input image despite all the aforementioned types of outliers, yielding a complete panorama for each input frame under the same exposure and illumination! Figure 6.1(e) and Figure 6.2 show some examples of our results.

In fact, the flexibility of the low-dimensional subspace model goes beyond merely handling the photometric variations. For instance, as one can see in Figure 6.2, the building facades on a street are never perfectly planar, and windows and doors do have some small depth variation. It causes small appearance changes across different views. In fact, those changes can also be well approximated by a low-dimensional subspace, and hence are retained in the panorama associated with each view in our results (see Figure 6.2(d) and the supplementary video). Different from occluding objects, keeping these small-depth variations actually produces more realistic presentation of the real scene than enforcing everything to lie on a single plane.

Remember that the first step for any accurate image stitching is to obtain pixel-wise pre-

cise image alignment. While image-intensity based (direct) methods are often used to align sequential frames in a video, they typically assume the intensity remains constant over time. Recently, using the same idea of matrix rank minimization, [85] proposes a novel method called *Robust Alignment by Sparse and Low-rank decomposition* (RASL) for aligning a batch of linearly correlated images. However, like other direct methods, RASL requires a reasonable initialization. In this chapter, we extend RASL to automatically register video sequences by initializing it using the robust plane detection and tracking algorithm TRANSAC introduced in Chapter 2. By combining the strengths of both intensity-based and feature-based methods, our new method achieves fully automatic alignment with pixel-wise accuracy.

6.1.1 Related Work

Low-dimensional subspace models

The low-dimensional subspace model has been extensively used to model appearances of objects (e.g., faces) as well as indoor and outdoor scenes in computer vision. For example, many empirical and theoretical results exist for Lambertian scenes [36, 10, 8]. Most notably, [8] has shown that the set of images of a convex Lambertian object obtained under arbitrary distant lighting sources lies close to a 9D linear subspace.

Recently, there has been increasing interest in studying the appearances of indoor and outdoor scenes under general lighting and viewing conditions [101, 45]. In particular, for man-made scenes with a small number of surface orientations, [45] derives upper bounds on the dimensionality of their appearances. Consider the pixel-wise aligned images $I_1, \ldots, I_n$ of a scene consisting of k_ρ different materials or BRDFs and k_n distinct surface normals obtained from arbitrary distant viewpoints under arbitrary distant illuminations. The main result of [45] says that, neglecting cast shadows, the rank of the matrix M formed by stacking $I_1, \ldots, I_n$ as columns is at most $k_\rho k_n$. This result suggests that for street view videos where $k_n = 1$, the rank of M is as low as the number of different BRDFs in the scene.

In addition to the above variations of the scene appearance, images of a scene may further undergo some global linear photometric transformations due to the changes of camera parameters (e.g., exposure, color balance) during the recording process, or the different response functions of different cameras [38, 101, 6]. For gray-level images $I_1, I_2 \in \mathbb{R}^m$, the most common transformation is the gain and bias model: $I_2 = aI_1 + b$ where a, b are two scalars. For color images, the affine transformation is often used to model the mixture of color channels, in addition to the gain and bias of each channel. For any two color images, the affine transformation can be written as

$$[I_{2,r}, I_{2,g}, I_{2,b}]^T = \mathcal{A}\,[I_{1,r}, I_{1,g}, I_{1,b}]^T + \boldsymbol{b}\mathbf{1}^T \tag{6.1}$$

where $\mathcal{A}$ is a 3×3 matrix, and $\boldsymbol{b}$ is a 3-vector. It is easy to see that images subject to such linear transformations also lie in a low-dimensional subspace.

Finally, the low-dimensional subspace model has also been successfully used to represent non-parametric motion fields [12, 31]. It is argued in [31] that, for videos with very small motions, the components and coefficients obtained through a PCA decomposition can be interpreted as a model of motion fields within the scene. This actually agrees well with our previous observation that small depth variations are kept in our low-rank panoramas.

Image stitching

The problem of image alignment has been extensively studied in the literature. Existing methods can be roughly classified into two categories. On one hand, direct alignment methods [71, 93] work on image regions and provide accurate registration using local algorithms, but need a good initialization. On the other hand, feature-based methods rely on detecting and matching a set of feature points, such as corners and SIFT features [96, 25, 15]. They do not require an initialization, but often fail to achieve pixel-wise registration accuracy even with global multi-frame bundle adjustment. Recent work on video registration has been

focused on the efficiency of feature-based methods [99]. In this work, we take advantage of both types of methods to achieve robust and accurate registration of video frames.

Given multiple aligned input images, there exist many works addressing the problem of composing a final mosaic image, such as pixel and seam selection, blending and exposure compensation. In particular, pixel and seam selection techniques aim to eliminate the ghost effects due to moving objects, and keep exactly one copy of each object in the final result, by using a minimum likelihood selection criterion [3], a weighted average in the regions of difference [109, 15], or interacting with users [2]. However, these methods are not always reliable in practice, as separating moving foregrounds from background for videos remains an open problem [24, 73].

To compensate for differences in exposure or illumination from the source images, many sophisticated blending algorithms have been developed in the literature. [15] uses pyramid blending to compensate for exposure differences, and [86] develops a gradient domain blending method to do seamless object insertion in image editing applications. Several variants of [86] with different cost functions have been studied in [62] to further improve its performance. Meanwhile, readers are referred to [116] for a comprehensive performance evaluation on existing color correction approaches. Finally, [35] proposes to convert each image into a radiance image using its exposure value and then create a stitched, high dynamic range image. While all the aforementioned methods focus on eliminating photometric variations from input images, it is the novelty of our method to directly model these variations using a low-dimensional subspace model and recover them all in the final results.

6.2 Problem Formulation

We begin introducing our method with a formal definition of the low-dimensional subspace model for video panoramas. Suppose we are given n complete and pixel-wise aligned panoramas (w.r.t. a common coordinate system) of a scene from different viewpoints under po-

tentially varying exposures or illuminations. We stack all the m pixels of each panorama as a vector, and denote them as $I_1^0, \ldots, I_n^0 \in \mathbb{R}^m$. If we put these vectors as columns of the matrix

$$A \doteq [I_1^0, \ldots, I_n^0] \in \mathbb{R}^{m \times n}, \tag{6.2}$$

then, according to our discussion in the previous section, the matrix A has a very low rank depending on the scene and camera. This is illustrated in Figure 6.3.[1]

For the image stitching problem, in each input image I_j we only see a deformed version of a very small portion of the entire scene. Particularly, a video sequence along a street can be thought as a sliding window through which each frame sees a small chunk of the street from a different view. If the camera is a perspective projection, then there exist homography matrices $\tau_1, \ldots, \tau_n \in \mathbb{GL}(3)^2$ which transform the input video frames $I_1, \ldots, I_n$ into a common coordinate system on the dominant plane, respectively.

In addition, since each image I_j $(1 \leq j \leq n)$ has a limited field of view and only sees a very small portion of the scene, there is an associated support Ω_j that indicates the observed region (entries) from the j-th view, as illustrated in Figure 6.3. We write $\mathcal{P}_{\Omega_j}(I_j)$ as the projection of I_j to the space of vectors supported on Ω_j. With a slight abuse of notation, we also use $\Omega_j \in \mathbb{R}^m$ as a vector to represent the observed pixels in I_j^0, where $\Omega_j(k) = 1$ if the k-th pixel in I_j^0 is observed, and $\Omega_j(k) = 0$ otherwise. Hence the video frames are related to the complete panoramas as

$$\mathcal{P}_{\Omega_j}(I_j \circ \tau_j) = \mathcal{P}_{\Omega_j}(I_j^0). \tag{6.3}$$

Given the transformation matrices $\tau = \{\tau_j\}_{j=1}^n$, we can write the aligned data matrix as $\mathcal{P}_\Omega(D \circ \tau) \doteq [\mathcal{P}_{\Omega_1}(I_1 \circ \tau_1), \ldots, \mathcal{P}_{\Omega_n}(I_n \circ \tau_n)]$, where $\Omega \doteq [\Omega_1, \ldots, \Omega_n]$ are the supports associated with all the views. Then the image stitching problem naturally reduces to the

[1]For color images, one can write the data matrix as $A = [I_{1,r}^0, I_{1,g}^0, I_{1,b}^0, \ldots, I_{n,r}^0, I_{n,g}^0, I_{n,b}^0] \in \mathbb{R}^{m \times 3n}$, which also has a low rank depending on the scene and camera. However, we stick to gray-level images in the chapter for the simplicity of presentation.

[2]Here, GL stands for General Linear. This class of transformations is able to represent distortion in a perspective image of a planar object.

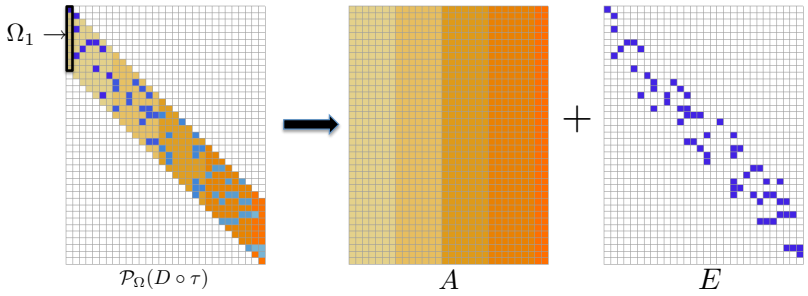


Figure 6.3: Illustration of problem formulation: Robust recovery of a low-rank matrix A from highly incomplete measurements within a band along the matrix diagonal.

following low-rank matrix completion problem:

$$\min_{\tau, A} \operatorname{rank}(A) \quad \text{s.t.} \quad \mathcal{P}_\Omega(D \circ \tau) = \mathcal{P}_\Omega(A). \tag{6.4}$$

In practice, the low-rank structure of the aligned images can be easily violated, due to the presence of reflections and occluding objects in the scene. Since these errors typically affect only a small fraction of all pixels in an image, we can model them as sparse errors whose nonzero entries can have large magnitude. Let $\boldsymbol{e}_j$ represent the error corresponding to the j-th frame: $\mathcal{P}_{\Omega_j}(I_j \circ \tau_j) = \mathcal{P}_{\Omega_j}(I_j^0 + \boldsymbol{e}_j)$, and let $E = [\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n]$ be the matrix with all the error vectors as columns. Then, to recover the low-rank panoramas $\{I_j^0\}_{j=1}^n$, we actually need to solve the following more challenging problem of recovering a low-rank matrix from highly incomplete *and* corrupted observations:

$$\min_{\tau, A, E} \operatorname{rank}(A) + \nu \|E\|_0 \quad \text{s.t.} \quad \mathcal{P}_\Omega(D \circ \tau) = \mathcal{P}_\Omega(A + E), \tag{6.5}$$

where the ℓ_0-norm $\|\cdot\|_0$ counts the number of nonzero entries of a matrix, and $\nu > 0$ is a parameter that trades off the rank of the solution versus the sparsity of the error.

To summarize, our goal is to recover a set of homographies $\tau_1, \ldots, \tau_n$ that align all the frames to a common world coordinate system as well as the corresponding view-dependent panoramas, by minimizing the rank of a matrix A which agrees with the aligned input images $\{I_j \circ \tau_i\}_{j=1}^n$ on the observed regions Ω, up to some sparse gross errors E. Notice that here

(τ, A, E) are all unknowns.

The rest of the chapter is organized as follows. In Section 6.3, by assuming that the correct homographies τ are given, we introduce an efficient and effective solution to (6.5) via convex programming. Then, to obtain the homographies τ for the sequence, we rely on a robust video frame alignment algorithm discussed in Section 6.4. We conduct experiments on both synthetic and real data to illustrate the performance of our method and compare with other state-of-the-art techniques in Section 6.5.

6.3 Robust Low-rank Panoramas via Convex Optimization

In this section, we show how to solve the problem (6.5) when the correct transformations τ are given. Note that even with τ given, the objective function of problem (6.5) is still highly combinatorial, which is in general NP-hard if we are looking for the global optimal solution. However, by the recent advances in convex optimization, we can replace the non-linear functions rank$(\cdot)$ and ℓ_0-norm by their corresponding convex surrogates, as proposed by the work of [18, 26]. Specifically, we replace rank$(\cdot)$ by the nuclear norm $\|\cdot\|_*$[3], and $\|\cdot\|_0$ by the $\|\cdot\|_1$ norm[4], which leads to the following convex optimization problem:

$$\min_{A,E} \|A\|_* + \lambda \|E\|_1 \quad \text{s.t.} \quad \mathcal{P}_\Omega(D \circ \tau) = \mathcal{P}_\Omega(A + E), \tag{6.6}$$

where $\lambda = 1/\sqrt{m}$ is a scalar weight. Since the pioneering work [92, 21], there have been extensive theoretical results that provide evidence for the effectiveness of using such convex surrogates for recovering sparse signals and low-rank matrices. In particular, [18, 63] have shown that in the case when Ω contains a constant fraction of the entries, the above convex program succeeds with high probability under mild conditions. In fact, this type of "low-

[3]Sum of all singular values of a matrix.
[4]Sum of absolute values of all entries.

dimension + sparse" model has been proven to be very powerful and widely used for image and video processing tasks such as denoising [57], segmentation [28] and compressive imaging [95].

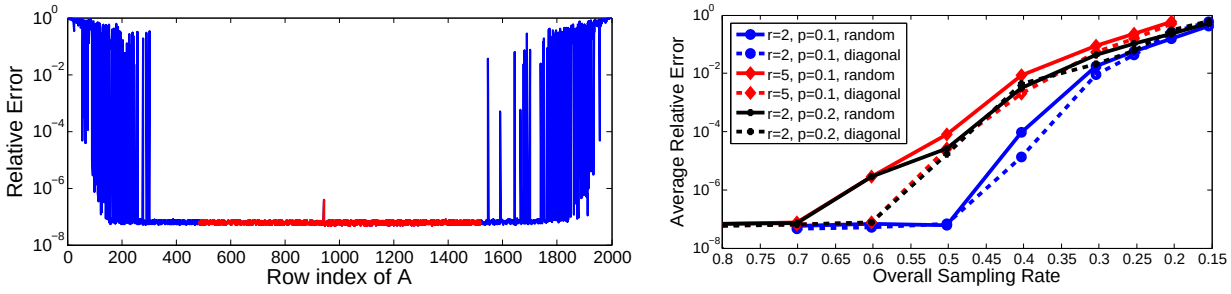
Similar recoverability results have also been obtained for a more general low-dimensional subspace Ω, known as SpaRCS [113] or compressive principal component pursuit [115] . In a nutshell, the results in [115] claim that, if (A_0, E_0) are incoherent, then the recovery from (6.6) is exact with high probability if "$\dim(Q) \geq C \cdot \log^2 n \times$ degrees of freedom(A_0, E_0)," where Q is a randomly chosen observable subspace according to the Haar measure. Curious readers are referred to [115] for the detailed proofs. Notice that if the matrix A has a fixed rank r – which is the case in our setting, then a lower-bound on the number of measurements needed is $O(rn \log^2 n)$ which is only a diminishing fraction of the entries $O(n^2)$ of the matrix as n goes to infinity. This is actually the case when the length of the video sequence grows large. In other words, the results of [115] suggest that as long as the resolution of the image frame – the size of the support $|\Omega_1|$ in Figure 6.3 – grows as $O(r \log^2 n)$, then good recovery of the low-rank panoramas and sparse errors from (6.6) is possible.[5]

Finally, many efficient and scalable first-order methods have been proposed in recently years to solve (6.6). In this work we adopt the alternating direction method developed in [103].

6.3.1 Simulation on Synthetic Data

While our formulation of the stitching problem is largely inspired by the robust PCA theory [18], there is a major difference between our problem and the conditions for exact recovery of a low-rank matrix using (6.6). Specifically, for the exact recovery property to hold, it is assumed in [18] that entries in Ω are selected uniformly at random among all entries of $D \circ \tau$. However, as one can see in Figure 6.3, the observed regions for our problem are within a

[5] In practice, the resolution of a frame is often fixed. So the existing theory actually cannot imply exact recovery for our stitching problem. However, experiment results show that our approach works well for fairly long videos. We consider bridging this gap between theory and practice as an interesting future direction.



(a) (b)

Figure 6.4: Simulation results. **(a)** Relative error in estimating each row of A. Rows in red correspond to the ones whose sampling rates are greater or equal to the overall sampling rate. **(b)** Average relative error for rows in S as a function of overall sampling rate.

band along the matrix diagonal. One particular problem with it is that image pixels near the boundary of the visible region in the panorama are sampled much less frequently than other pixels.

To understand the effect of diagonal sampling on the solution to (6.6), we first conduct some simulations on synthetic data. The observation matrix $M \in \mathbb{R}^{m \times n} = \mathcal{P}_\Omega(A_0 + E_0)$ is generated as follows: First, we generate a rank-r matrix A_0 as a product $A_0 = XY^T$ where X and Y are random matrices of size $m \times r$ and $n \times r$, respectively, with entries independently sampled from a $\mathcal{N}(0, 1)$ distribution. Then, E_0 is generated by choosing a support set of size $p \times m \times n$ uniformly at random, and assigning each of its non-zero entries a value independently sampled from the uniform distribution in $[-5, 5]$. Finally, to generate the diagonal sampling pattern Ω, for the j-th column we set $\Omega_j(k) = 1$ if $(j-1) \times q < k \leq m - (n-j) \times q$, and $\Omega_j(k) = 0$ otherwise. Here, q controls the width of the band along the matrix diagonal.

In the first experiment, we fix $m = 2000, n = 200, r = 2, p = 0.1$ and $q = 6$. Let (A^*, E^*) be the estimated low-rank and sparse matrices from M by solving (6.6). Figure 6.4(a) shows the relative error in estimating each row of A, which is defined as $\epsilon_k = \frac{\|A_k^* - A_{0,k}\|_2}{\|A_{0,k}\|_2}, 1 \leq k \leq m$, where A_k^* and $A_{0,k}$ are the k-th row of A^* and A_0, respectively. As one can see, rows at the two ends of the matrix have very large errors. This is expected because there are simply not

(a) (b)

Figure 6.5: Handling non-uniform sampling. **(a)** Panorama obtained by solving (6.6) for the sequence shown in Figure 6.1. White curve represents the boundary of the panorama. Note that regions close to the boundary are darker than other regions. **(b)** Panorama obtained by solving (6.7). Notice the improvement at pixels near the boundary.

enough samples for these rows.

More importantly, one can also see in Figure 6.4(a) that accurate recovery of the low-rank matrix A_0 is possible for rows in the middle with sufficient number of samples, despite a constant fraction of corrupted entries. To further understand this phenomenon, we define the overall sampling rate of the data matrix as $\alpha = \frac{\sum_{j,k} \Omega_j(k)}{m \times n}$; then let S be the set of rows whose sampling rates are greater or equal to the overall sampling rate: $S = \{k \in \{1, 2, \ldots, m\} : \sum_{j=1}^{n} \Omega_j(k)/n \geq \alpha\}$, as shown in Figure 6.4(a).

Now, fixing $m = 2000, n = 200$, we vary the overall sampling rate by changing the value of q. For each q, we generate a data matrix M and obtain (A^*, E^*) by solving (6.6). We compute the average relative error for all rows in S: $\epsilon_S = \sum_{k \in S} \epsilon_k / |S|$ and plot it as a function of α in Figure 6.4(b), for different values of r and p.[6] As one can see, the error remains very small for fairly small sampling rates. For example, for $\alpha = 0.3$, ϵ_S is less than 0.1 for all cases.

In addition, in Figure 6.4(b) we also plot the average relative error when the entries in Ω are selected uniformly at random with probability α, as suggested by the robust PCA theory [18]. We can see that the performance of the convex program (6.6) on data matrices with diagonal-band sampling (restricted to rows in S) is as good as on those with random sampling for the same sampling rate α!

[6] The result is averaged over 10 trials for each choices of r, p and q.

6.3.2 Handling Non-uniform Sampling for Image Stitching

For real image stitching problem, the same error pattern occurs as the synthetic data case. This is evidenced in Figure 6.5(a), which shows the estimated panorama by solving (6.6). As one can see, the regions close to the boundary of the panorama are darker than other regions. This suggests that the corresponding rows of A^* tend to have zero values.

To remedy this problem, we propose to use a different weight for each row of E in (6.6), instead of a single scalar λ. Intuitively, we wish to incur a heavier penalty for non-zero entries of E in the rows with fewer samples. Therefore, for the k-th row, we set $\lambda_k = \max\{1, 0.5n/\sum_{j=1}^{n} \Omega_j(k)\}/\sqrt{m}$ and propose to solve the following optimization problem:

$$\min_{A,E} \|A\|_* + \|\Lambda E\|_1 \quad \text{s.t.} \quad \mathcal{P}_\Omega(D \circ \tau) = \mathcal{P}_\Omega(A + E), \tag{6.7}$$

where $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_m) \in \mathbb{R}^{m \times m}$ is a diagonal matrix. The panorama obtained by solving (6.7) is shown in Figure 6.5(b), where the missing pixels near the boundary are now recovered. However, as we will see later (e.g., Figure 7), by setting a large penalty to regions near the boundary, one actually compromises the robustness to outliers in these regions. This is often acceptable as there are simply not enough samples to separate low-rank component from sparse errors in these regions.

6.4 Robust and Accurate Video Registration

We have seen from previous sections that by imposing low-rankness on the desired solution, one can robustly and efficiently recover the complete panoramas of the scene despite gross errors. In fact, with some proper modifications, the same idea of matrix rank minimization can be used to obtain accurate estimates of the homography matrices among all video frames. Specifically, a new direct alignment algorithm called *Robust Alignment by Sparse and Low-rank decomposition* (RASL) has been recently proposed by [85] to register a batch of linearly

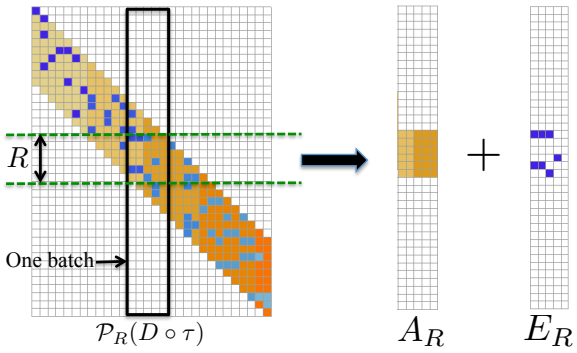


Figure 6.6: Illustration of robust alignment of a consecutive batch of frames with a common overlap region R.

correlated images. In this section, we briefly discuss how to apply RASL to our problem of aligning video frames, resulting in a fully automatic image stitching system for low-rank panoramas.

Given n input images $\{I_j\}_{j=1}^n$, recall that the ideal observation model is $\mathcal{P}_{\Omega_j}(I_j \circ \tau_j) = \mathcal{P}_{\Omega_j}(I_j^0)$ for the j-th image. Denote $R = \bigcap_{j=1}^n \Omega_j$ as the intersection of observed regions among images considered – see Figure 6.6 for an illustration. If R is *not* empty, we write $\mathcal{P}_R(I_j \circ \tau_j), j = 1, \ldots, n$ as the projection of the aligned input images to the space of vectors supported on R. Then, if we stack each $\mathcal{P}_R(I_j \circ \tau_j)$ as column of a new data matrix:

$$\mathcal{P}_R(D \circ \tau) = [\mathcal{P}_R(I_1 \circ \tau_1), \ldots, \mathcal{P}_R(I_n \circ \tau_n)], \tag{6.8}$$

this matrix should also have a very low rank. In the presence of errors, we can write our observation model as:

$$\mathcal{P}_R(D \circ \tau) = A_R + E_R, \tag{6.9}$$

where A_R, E_R represent the low-rank component and sparse error component, respectively, with their k-th entries being zero for any $k \notin R$. Then, using the same argument as in Section 6.2, we can cast the problem of joint image alignment as the following optimization problem:

$$\min_{A_R, E_R, \tau} \|A_R\|_* + \lambda \|E_R\|_1 \quad \text{s.t.} \quad \mathcal{P}_R(D \circ \tau) = A_R + E_R. \tag{6.10}$$

Algorithm 2 (Robust Video Registration via RASL)

1: **Input:** Input images $\{I_j\}_{j=1}^n$, number of batches L and batch size p.
2: $\tau_1^* \leftarrow I_{3\times 3}$.
3: **for** the i-th batch:
4: Set $D^l \leftarrow [I_{p\times(l-1)+1}, \ldots, I_{p\times l+1}]$; $R^l \leftarrow \bigcap_{j=p\times(l-1)+1}^{p\times l+1} \Omega_j$.
5: Compute $\tau^l = [\tau_{p\times(l-1)+1}, \ldots, \tau_{p\times l+1}]$ by solving the following RASL problem:

$$\min_{A,E,\tau^l} \|A\|_* + \lambda\|E\|_1 \quad \text{s.t.} \quad \mathcal{P}_{R^l}(D^l \circ \tau^l) = A + E.$$

6: Update $\tau_j^* \leftarrow \tau_j \cdot \tau_{p\times(l-1)+1}^{-1} \cdot \tau_{p\times(l-1)+1}^*$ for $p \times (l-1) + 2 \le j \le p \times l + 1$.
7: **end for**
8: **Output:** A set of homographies $\tau_1^*, \ldots, \tau_n^*$.

In [85], given a good initialization, (6.10) is solved by iteratively linearizing the nonlinear equality constraint at the current estimate of τ, yielding a sequence of convex programs whose solutions converge quadratically to the correct alignment. It has been shown in [85] that RASL is able to achieve pixel-wise alignment accuracy over a wide range of realistic misalignments and corruptions.

However, in order to apply the above scheme to street view video sequences, there are two important issues which need to be addressed. First, for a typical long video sequence, there is often no common region among all frames. To deal with this problem, we divide the entire sequence into L multiple small (overlap) batches of size $(p+1)$, so that the l-th batch contains frames $(p \times (l-1) + 1)$ to $(p \times l + 1)$, and apply RASL to solve (6.10) for each batch individually. Note that the way we divide the sequence ensures that any two adjacent batches share exactly one frame, which enables us to link all the transformations between frames into a common coordinate system in the end. In addition, as suggested by [85], the value of p should be chosen as large as possible, as the low-rank model works better when p is much larger than the dimension of the subspace spanned by the intrinsic views. In our problem, however, p is restricted by the condition that R must be large enough so that (6.10) can be solved reliably in the presence of gross errors. See Figure 6.6 for an illustration of the relation between the size of R and the batch size p. In this work, we fix $p = 10$ for

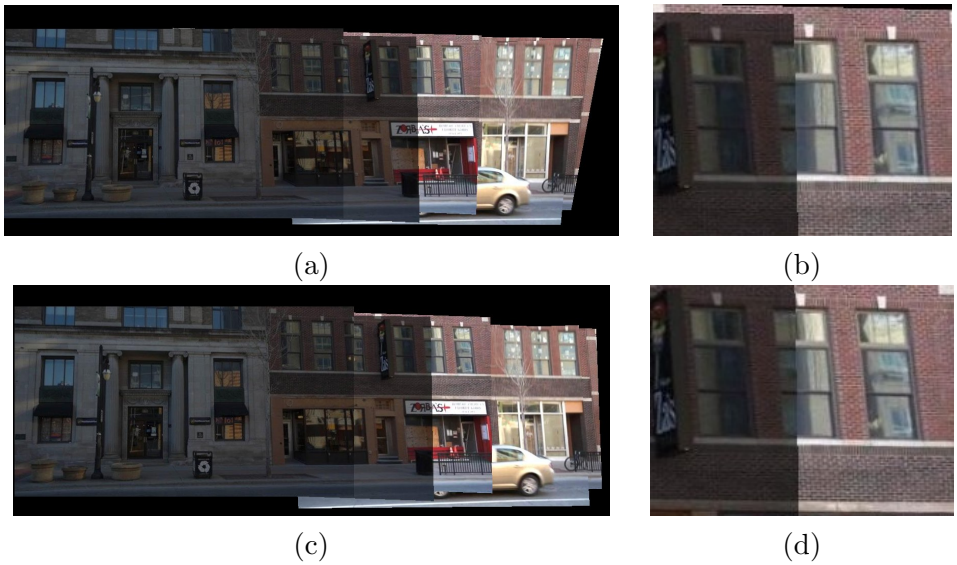


Figure 6.7: Pixel-wise accurate alignment via matrix rank minimization. **(a)** Overlapping result according to the homographies estimated by the feature-based plane detection and tracking algorithm [54]. **(c)** Overlapping result according to the homographies refined by RASL. The improvement of alignment quality is clear. **(b) and (d)** Details of a small region in (a) and (c), respectively.

videos taken by our own and $p = 8$ for Google Map Street View sequences.

Second, as a local method, a good initialization of the transformation parameters τ is needed. For this purpose, one may simply use the standard point-based RANSAC algorithm for two-view homography estimation [54]. However, in this chapter we adopt the TRANSAC algorithm developed in Chapter 2. It generalizes two-view RANSAC to estimate consistent plane models across multiple frames, and has a very high breakdown point to gross outliers. Finally, we summarize our robust video registration algorithm in Algorithm 2.

In Figure 6.7(a), we show the alignment result obtained by the plane detection and tracking algorithm. As one can see in Figure 6.7(b), due to the fact that feature point localization is often noisy, the feature-based method cannot achieve pixel-wise alignment accuracy, even with multi-view bundle adjustment. We note that more instances of the same problem can be found in the panoramas obtained by AutoStitch (Figure 6.9(b)), which also uses a feature-based method for alignment. On the contrary, using the output of featur-based method as initialization, a direct method such as RASL can greatly improve the alignment

quality, as shown in Figure 6.7(c) and (d), as well as in Figure 6.9(a).

6.5 Experiments

In this section, we report results of our method on both videos captured by ourselves using a hand-held SONY NEX-5N camera (Figure 6.8) and videos from the Google Map Street View database[7] captured by camera mounted on a moving car (Figure 6.10). Note that for videos taken by our own, we use the newly developed camera calibration system [120] to remove radial distortion.

To better understand the advantages of our method, we compare our method against the state-of-the-art image stitching system AutoStitch [15], and the popular software Photomerge in Adobe Photoshop CS5[8], which is largely based on the work of [2, 1].

Our Video Sequences. In Figure 6.9, we show the results of both methods on sequence H1 to H3. As one can see, our method performs consistently better than AutoStitch and Photomerge, producing clean, pleasing-looking results with pixel-wise registration accuracy. In particular, comparing Figure 6.9(a) with the corresponding input images, one can see that objects do not belong to the dominant plane (e.g., cars, trees) have been completely removed from the panoramas by our method, except for some small regions close to the boundary due to insufficient number of samples as we discussed before. For example, see the lower-left corner of the panorama for sequence H1 in Figure 6.9(a). On the contrary, Both AutoStitch and Photomerge perform poorly in removing outliers, resulting in significant ghosting effects (Figure 6.9(b)) or cutting-off effects (Figure 6.9(c)). In addition, Photomerge also has difficulties in matching video frames, possibly due to the repetitive patterns.

Google Map Street View Sequences. Next, we compare our method with AutoStitch on the Google Map Street View database (Figure 6.10). As one can see in Figure 6.11, our method works very well on both sequences, while AutoStitch have obvious problems

[7] maps.google.com/streetview

[8] http://www.adobe.com/products/photoshop.html

Figure 6.8: Snapshots of testing videos taken by a hand-held camera. **From top to bottom:** Sequences H1, H2, and H3. The number of frames for each sequence is 101, 81 and 121, respectively.

(a) Our method (b) AutoStitch

(c) Photomerge

Figure 6.9: Comparison of video stitching results on sequences H1 to H3.

registering the input images. Furthermore, our method successfully remove outliers, such as pedestrians, reflections on the window, and signs on the ground, from the panoramas, while preserving details on the dominant planes.

G1

G2

Figure 6.10: Snapshots of testing videos from Google Map Street View database. The number of frames for both sequences is 57.

(a) Our method

(b) AutoStitch

Figure 6.11: Video stitching results on Google Map Street View sequences G1 and G2.

Discussion. From the above results, we have seen that by harnessing the intrinsic relationships across multiple frames of a street view video via a low-dimensional subspace model, we can generate panoramas for the video frames in a new holistic and effective way, despite exposure and illumination differences, parallax, reflections, and occluding objects. From comparison with conventional image stitching methods, one may also notice that pixels of our panoramas are all "estimated", not "stitched" together from raw pixels of original images. As result, our panoramas are not as sharp in local regions, but their global appearance is much more clean and homogeneous and their global geometry is much more accurate and consistent. This opens up good future research directions on how to combine this new method with conventional stitching methods to achieve both local sharpness and global accuracy for the final panoramas.

Chapter 7

Discussion and Conclusions

In this thesis, we have demonstrated the power of exploring structural regularities in solving challenging problems in 3D reconstruction and modeling of urban scenes from images and videos. In particular, we have developed a series of tools to exploit different types of structural regularities, including planar surfaces, repetitive patterns, symmetries and the linear correlations across multiple images. In Chapters 2 and 3, we described a novel structure from motion technique using one or more large planes in the scene, and further demonstrated how the reconstructed planes can be used to substantially improve the performance of existing video stabilization methods. In Chapters 4 – 6, we introduced a recently developed technique for robust low-rank matrix recovery, and applied it to reconstruct geometric and textural models of urban scenes from both large-baseline images and long video sequences.

Indeed, with the fast development of acquisition technology, we now have access to a huge amount of visual data from street view videos to aerial photos for city-scale 3D modeling. Clearly, there is still a gap between the results one can get from existing techniques and the desired high-quality city-scale 3D models for industrial applications. Here, we believe that the main difficulties lie in the scale of the collected visual data, as well the inherent complexity in the geometry and topology of urban scenes. In the rest of this chapter, we outline a few promising directions for future work that would strengthen the current systems for 3D reconstruction and modeling, as well as influence many other applications in computer vision.

Accurate piecewise planar scene segmentation. A crucial step towards urban scene understanding is to segment images into geometrically consistent regions (e.g., building fa-

cades), often under severe occlusions, and identify the precise boundaries between them. However, none of the existing methods, including those developed in this thesis, can provide satisfactory solutions to this problem. For example, the segmentation algorithm we introduced in Chapter 5 can obtain accurate boundaries between two building facades, but is very slow, and can only handle simple cases where each image is dominated by one or two large planar regions. Meanwhile, the piecewise planar and non-planar scene segmentation algorithm we developed in Chapter 3 is very efficient, and can handle more complicated scene geometry. However, it cannot get accurate segmentation boundaries. Therefore, we believe that it is necessary to combine local and global features within each frame, as well as the temporal correlations among multiple frames, in order to achieve the desired results.

Handling complex scene geometry. In practice, urban scenes may contain many objects whose geometry is more sophisticated than the few planar surfaces currently assumed by our methods. For example, in a recent paper [119], TILT has been extended to reconstruct another class of surfaces called the generalized cylindrical surfaces, which are very common in urban environments. Meanwhile, non-planar repeating structures, such as the balconies of a building, are also extremely important for modeling urban scenes. Therefore, we need to extend our method to handle more complex parametric deformation models, and develop new computational tools which can exploit the underlying structural regularities with respect to such models.

Fast and scalable algorithms. Another important aspect of any practical algorithm is its efficiency. In this thesis, we have already seen a few instances of discovering the structural regularities via new powerful computational tools such as graph cuts and first-order methods for low-rank matrix recovery. However, in practice, these methods are still not fast enough for time-critical applications on very large datasets. For example, the holistic segmentation algorithm based on low-rank textures currently requires a large number of SVD computations for each image patch. It remains an open question whether or not there is an alternative approach to the fast recovery of low-rank textures from images.

References

[1] A. Agarwala. Efficient gradient-domain compositing using quadtrees. *ACM Transactions on Graphics*, 26(3):94, 2007.

[2] A. Agarwala, M. Agrawala, M. F. Cohen, D. Salesin, and R. Szeliski. Photographing long scenes with multi-viewpoint panoramas. *ACM Transactions on Graphics*, 25(3):853–861, 2006.

[3] A. Agarwala, M. Dontcheva, M. Agrawala, S. M. Drucker, A. Colburn, B. Curless, D. Salesin, and M. F. Cohen. Interactive digital photomontage. *ACM Transactions on Graphics*, 23(3):294–302, 2004.

[4] C. Baillard and A. Zisserman. Automatic reconstruction of piecewise planar models from multiple views. In *CVPR*, 1999.

[5] A. Bartoli. A random sampling strategy for piecewise planar scene segmentation. *Computer Vision and Image Understanding*, 105(1):42–59, 2007.

[6] A. Bartoli. Groupwise geometric and photometric direct image registration. *Proceedings of the IEEE*, 30(12):2098–2108, 2008.

[7] A. Bartoli and P. Strum. Constrained struture and motion from multiple uncalibrated views of a piecewise planar scene. *International Journal of Computer Vision*, 52(1):45–64, 2003.

[8] R. Basri and D. Jacobs. Lambertian reflectance and linear subspaces. *Proceedings of the IEEE*, 25(2):218–233, 2003.

[9] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded up robust features. *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[10] P. N. Belhumeur and D. J. Kriegman. What is the set of images of an object under all possible illumination conditions? *International Journal of Computer Vision*, 28(3):245–260, 1998.

[11] P. Bhat, C. L. Zitnick, N. Snavely, A. Agarwala, M. Agrawala, M. F. Cohen, B. Curless, and S. B. Kang. Using photographs to enhance videos of a static scene. In *Rendering Techniques*, 2007.

[12] M. J. Black, Y. Yacoob, A. D. Jepson, and D. J. Fleet. Learning parameterized models of image motion. In *CVPR*, 1997.

[13] B. Bocquillon, P. Gurdjos, and A. Crouzil. Towards a guaranteed solution to plane-based self-calibration. In *ACCV*, 2006.

[14] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239, 2001.

[15] M. Brown and D. G. Lowe. Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, 74(1):59–73, 2007.

[16] A. M. Bruckstein, D. L. Donoho, and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Review*, 51(1):34–81, 2009.

[17] C. Buehler, M. Bosse, and L. McMillan. Non-metric image-based rendering for video stabilization. In *CVPR*, 2001.

[18] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58(3), 2011.

[19] E. J. Candès and Y. Plan. Tight oracle bounds for low-rank matrix recovery from a minimal number of random measurements. *Preprint*, 2009.

[20] E. J. Candès and Y. Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010.

[21] E. J. Candès and B. Recht. Exact matrix completion via convex optimzation. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

[22] E. J. Candès, J. K. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Comm. Pure Appl. Math*, 59(8):1207–1223, 2006.

[23] E. J. Candès and T. Tao. Decoding by linear programming. *IEEE Trans. Inform. Theory*, 51(12):4203–4215, 2005.

[24] V. Cehver, M. F. Duarte, C. Hedge, and R. R. Baraniuk. Sparse signal reocvery using Markov random fields. In *NIPS*, 2008.

[25] T. J. Cham and R. Cipolla. A statistical framework for long-range feature matching in uncalibrated image mosaicing. In *CVPR*, 1998.

[26] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572–596, 2011.

[27] B. Chen, K. Lee, W. Huang, and J. Lin. Capturing intention-based full-frame video stabilization. *Comput. Graph. Forum*, 27(7):1805–1814, 2008.

[28] B. Cheng, G. Liu, J. Wang, Z. Huang, and S. Yan. Multi-task low-rank affinity pursuit for image segmentation. In *ICCV*, 2011.

[29] O. Chum, T. Werner, and J. Matas. Two-view geometry estimation unaffected by a dominant plane. In *CVPR*, 2005.

[30] D. Crandall, A. Owens, N. Snavely, and D. Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In *CVPR*, 2011.

[31] M. Dixon, A. Abrams, N. Jacobs, and R. Pless. On analyzing video with very small motions. In *CVPR*, 2011.

[32] D. L. Donoho. For most large underdetermined systems of linear equations the minimal ℓ_1-norm solution is also the sparsest solution. *Comm. Pure Appl. Math*, 59:797–829, 2004.

[33] D. L. Donoho, M. Elad, and V. N. Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Trans. Inform. Theory*, 52(1):6–18, 2006.

[34] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1:211–218, 1936.

[35] A. Eden, M. Uyttendaele, and R. Szeliski. Seamless image stitching of scenes with large motions and exposure differences. In *CVPR*, 2006.

[36] R. Epstein, P. W. Hallinan, and A. L. Yuille. 5+/-2 eigenimages suffice: An empirical investigation of low-dimensional lighting models. In *IEEE Workshop on Physics-Based Modeling in Computer Vision*, 1995.

[37] I. Esteban, J. Dijk, and F. C. A. Groen. FIT3D toolbox: multiple view geometry and 3d reconstruction for Matlab. In *International Symposium on Security and Defence Europe (SPIE)*, 2010.

[38] G. Finlayson, M. Drew, and B. Funt. Color constancy: Generalized diagonal transforms suffice. *J. Optical Society of America A*, 11(11):3011–3019, 1994.

[39] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[40] A. W. Fitzgibbon, Y. Wexler, and A. Zisserman. Image-based rendering using image-based priors. *International Journal of Computer Vision*, 63(2):141–151, 2005.

[41] A. W. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed or open image sequences. In *ECCV*, 1998.

[42] J. Frahm, P. Fite-Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y. Jen, E. Dunn, B. Clipp, S. Lazebnik, and M. Pollefeys. Building Rome on a cloudless day. In *ECCV*, 2010.

[43] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Manhattan-world stereo. In *CVPR*, 2009.

[44] D. Gallup, J. Frahm, and M. Pollefeys. Piecewise planar and non-planar stereo for urban scene reconstruction. In *CVPR*, 2010.

[45] R. Garg, H. Du, S. M. Seitz, and N. Snavely. The dimensionality of scene appearance. In *ICCV*, 2009.

[46] R. Gherardi and A. Fusiello. Practical autocalibration. In *ECCV*, 2010.

[47] M. Gleicher and F. Liu. Re-cinematography: Improving the camerawork of casual video. *TOMCCAP*, 5(1), 2008.

[48] A. Goldstein and R. Fattal. Video stabilization using epipolar geometry. *ACM Transactions on Graphics*, 32(5), 2012.

[49] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *ICCV*, 2009.

[50] M. Grundmann, V. Kwatra, D. Castro, and I. Essa. Effective calibration free rolling shutter removal. *IEEE ICCP*, 2012.

[51] M. Grundmann, V. Kwatra, and I. A. Essa. Auto-directed video stabilization with robust l1 optimal camera paths. In *CVPR*, 2011.

[52] A. Gupta, A. Efros, and M. Hebert. Blocks world revisited: image understanding using qualitative geometry and mechanics. In *ECCV*, 2010.

[53] P. Gurdjos and P. Sturm. Methods and geometry for plane-based self-calibration. In *CVPR*, 2003.

[54] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, 2000.

[55] K. Huang, A. Yang, W. Hong, and Y. Ma. Large baseline matching and reconstruction from symmetry cells. In *ICRA*, 2004.

[56] T. Igarashi, T. Moscovich, and J. F. Hughes. As-rigid-as-possible shape manipulation. *ACM Transactions on Graphics*, 24(3):1134–1141, 2005.

[57] H. Ji, C. Liu, Z. Shen, and Y. Xu. Robust video denoising using low rank matrix completion. In *CVPR*, 2010.

[58] I. Jolliffe. *Principal Component Analysis*. Springer-Verlag, 1986.

[59] T. Korah and C. Rasmussen. Analysis of building textures for reconstructing partially occluded facades. In *ECCV*, 2008.

[60] D. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. In *CVPR*, 2009.

[61] K. Lee, Y. Chuang, B. Chen, and M. Ouhyoung. Video stabilization using robust feature trajectories. In *ICCV*, 2009.

[62] A. Levin, A. Zomet, S. Peleg, and Y. Weiss. Seamless image stitching in the gradient domain. In *ECCV*, 2004.

[63] X. Li. Compressed sensing and matrix completion with constant proportion of corruptions. *Preprint*, 2011.

[64] X. Liang, X. Ren, Z. Zhang, and Y. Ma. Repairing sparse low-rank texture. In *ECCV*, 2012.

[65] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen, and Y. Ma. Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix. In *CAMSAP*, 2009.

[66] F. Liu, M. Gleicher, H. Jin, and A. Agarwala. Content-preserving warps for 3d video stabilization. *ACM Transactions on Graphics*, 28(3), 2009.

[67] F. Liu, M. Gleicher, J. Wang, H. Jin, and A. Agarwala. Subspace video stabilization. *ACM Transactions on Graphics*, 30(1), 2011.

[68] S. Liu, Y. Wang, L. Yuan, J. Bu, P. Tan, and J. Sun. Video stabilization with a depth camera. In *CVPR*, 2012.

[69] Y. Liu, H. Hel-Or, C. Kaplan, and L. Van Gool. Computational symmetry in computer vision and computer graphics. *FTCGV*, 5:1–197, 2010.

[70] D. Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999.

[71] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, 1981.

[72] Y. Ma, J. Košecká, S. Soatto, and S. Sastry. *An Invitation to 3-D Vision, From Images to Models*. Springer-Verlag, New York, 2004.

[73] J. Mairal, R. Jenatton, G. Obozinski, and F. Bach. Network flow algorithms for structured sparsity. In *NIPS*, 2010.

[74] E. Malis and R. Cipolla. Camera self-calibration from unknown planar structures enforcing the multi-view constraints between collineations. *Proceedings of the IEEE*, 24(9):1268–1272, 2002.

[75] J. Matas, O. Chum, M. Urba, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *BMVC*, 2002.

[76] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H. Shum. Full-frame video stabilization with motion inpainting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(7):1150–1163, 2006.

[77] J. F. Menudet, J. M. Becker, T. Fournel, and C. Mennessier. Plane-based camera self-calibration by metric rectification of images. *Image Vision Comput.*, 26:913–934, July 2008.

[78] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1–2):43–72, 2005.

[79] B. Mičušík and J. Košecká. Piecewise planar city 3D modeling from street view panaramic sequences. In *ICCV*, 2009.

[80] B. Mičušík and J. Košecká. Multi-view superpixel stereo in urban environments. *International Journal of Computer Vision*, 89(1):106–119, 2010.

[81] B. Mičušík, H. Wildenauer, and J. Košecká. Detection and matching of rectilinear structures. In *CVPR*, 2008.

[82] S. Negahban, P. Ravikumar, M. J. Wainwright, and B. Yu. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. In *NIPS*, 2009.

[83] R. K. Nicolas, N. Dano, and R. Hartley. Plane-based projective reconstruction. In *ICCV*, 2001.

[84] M. Park, K. Brocklehurst, R. T. Collins, and Y. Liu. Deformed lattice detection in real-world images using mean-shift belief propagation. *Proceedings of the IEEE*, 31(10):1804–1816, 2009.

[85] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma. RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images. In *CVPR*, 2010.

[86] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Transactions on Graphics*, 22(3):313–318, 2003.

[87] M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a hand-held camera. *International Journal of Computer Vision*, 59(3):207–232, 2004.

[88] M. Pollefeys, D. Nisté, J.-M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S.-J. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewéius, R. Yang, G. Welch, and H. Towles. Detailed real-time urban 3d reconstruction from video. *International Journal of Computer Vision*, 78:143–167, 2008.

[89] M. Pollefeys, F. Verbiest, and L. Van Gool. Surviving dominant planes in uncalibrated structure and motion recovery. In *ECCV*, 2002.

[90] J. Prankl, M. Zillich, B. Leibe, and M. Vincze. Incremental model selection for detection and tracking of planar surfaces. In *BMVC*, 2010.

[91] S. Rao, H. Mobahi, A. Yang, S. Sastry, and Y. Ma. Natural image segmentation with adaptive texture and boundary encoding. In *ACCV*, 2009.

[92] B. Recht, M. Fazel, and P. A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, 2010.

[93] H. S. Sawhney and R. Kumar. True multi-image alignment and its application to mosaicing and lens distortion correction. *Proceedings of the IEEE*, 21(3):235–243, 1999.

[94] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(5):824–840, 2009.

[95] X. Shu and N. Ahuja. Imaging via three-dimensional compressive sampling (3DCS). In *ICCV*, 2011.

[96] H. Shum and R. Szeliski. Construction of panoramic image mosaics with global and local alignment. *International Journal of Computer Vision*, 36(2):101–130, 2000.

[97] S. N. Sinha, D. Steedly, and R. Szeliski. Piecewise planar stereo for image-based rendering. In *ICCV*, 2009.

[98] N. Snavely, S. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80:189–210, 2008.

[99] D. Steedly, C. Pal, and R. Szeliski. Efficiently registering video into panoramic mosaics. In *ICCV*, 2005.

[100] G. W. Stewart. Perturbation theory for the singular value decomposition. *SVD and Signal Processing, II: Algorithms, Analysis and Applications*, pages 99–109, 1991.

[101] K. Sunkavalli, F. Romeiro, W. Matusik, T. Zickler, and H. Pfister. What do color changes reveal about an outdoor scene? In *CVPR*, 2008.

[102] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. F. Tappen, and C. Rother. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(6):1068–1080, 2008.

[103] M. Tao and X. Yuan. Recovering low-rank and sparse components of matrices from incomplete and noisy observations. *SIAM Journal on Optimization*, 21(1):57–81, 2011.

[104] R. Toldo and A. Fusiello. Robust multiple structures estimation with j-linkage. In *ECCV*, 2008.

[105] R. Toldo and A. Fusiello. Photo-consistent planar patches from unstructured cloud of points. In *ECCV*, 2010.

[106] P. Torr. Geometric motion segmentation and model selection. *Philosophical Transactions of the Royal Society of London*, 356(1740):1321–1340, 1998.

[107] P. Torr, A. W. Fitzgibbon, and A. Zisserman. The problem of degeneracy in structure and motion recovery from uncalibrated image sequences. *International Journal of Computer Vision*, 32(1):27–44, 1999.

[108] B. Triggs. Autocalibration from planar scenes. In *ECCV*, 1998.

[109] M. Uyttendaele, A. Eden, and R. Szeliski. Eliminating ghosting and exposure artifacts in image mosaics. In *CVPR*, 2001.

[110] J. Čech, J. Matas, and M. Perdoch. Efficient sequential correspondence selection by cosegmentation. *Proceedings of the IEEE*, 32(9):1568–1581, 2010.

[111] A. Vedaldi and S. Soatto. Local features, all grown up. In *CVPR*, 2006.

[112] J. Y. A. Wang and E. H. Adelson. Representing moving images with layers. *IEEE Transactions on Image Processing*, 3(5):625–638, 1994.

[113] A. Waters, A. Sankaranarayanan, and R. Baraniuk. SpaRCS: Recovering low-rank and sparse matrices from compressive measurements. In *NIPS*, 2011.

[114] T. Werner and A. Zisserman. New techniques for automated architectural reconstruction from photographs. In *ECCV*, 2002.

[115] J. Wright, A. Ganesh, K. Min, and Y. Ma. Compressive principal component pursuit. In *ISIT*, 2012.

[116] W. Xu and J. Mulligan. Performance evaluation of color correction approaches for automatic multi-view image and video stitching. In *CVPR*, 2010.

[117] G. Zhang, X. Qin, W. Hua, T. Wong, P. Heng, and H. Bao. Robust metric reconstruction from challenging video sequences. In *CVPR*, 2007.

[118] Z. Zhang, X. Liang, A. Ganesh, and Y. Ma. TILT: Transform invariant low-rank textures. In *ACCV*, 2010.

[119] Z. Zhang, X. Liang, and Y. Ma. Unwrapping low-rank textures on generalized cylindrical surfaces. In *ICCV*, 2011.

[120] Z. Zhang, Y. Matsushita, and Y. Ma. Camera calibration with lens distortion from low-rank textures. In *CVPR*, 2011.

[121] S. Zhu and D. Mumford. A stochastic grammar of images. *FTCGV*, 2(4):259–362, 2006.

[122] M. Zuliani, C. S. Kenney, and B. S. Manjunath. The multiransac algorithm and its application to detect planar homographies. In *ICIP*, 2005.